

Ohjelmoinnin opetus matematiikan opetuksessa

Helsingin yliopisto
Matemaattis-luonnontieteellinen
tiedekunta
Matematiikan ja tilastotieteen lai-
tos
Matematiikan aineenopettaja
Pro gradu -tutkielma

Joulukuu 2019
Jaakko Kokkonen

Ohjaaja: Sirkka-Liisa Eriksson



Tiedekunta - Fakultet - Faculty Matemaattis-luonnontieteellinen		Laitos - Institution - Department Matematiikan ja tilastotieteen laitos	
Tekijä - Författare - Author Jaakko Kokkonen			
Työn nimi - Arbetets titel Ohjelmoinnin opetus matematiikan opetuksessa			
Oppiaine - Läroämne - Subject Matematiikan aineenopettaja			
Työn laji/ Ohjaaja - Arbetets art/Handledare - Level/Instructor Pro gradu -tutkielma / Sirkka-Liisa Eriksson		Aika - Datum - Month and year joulukuu 2019	Sivumäärä - Sidoantal - Number of pages 67 s + 3 liites.
Tiivistelmä - Referat - Abstract <p><i>Tavoitteet.</i> Tutkielmassa käydään läpi uusimman, vuonna 2014 julkaistun ja vuonna 2016 käyttöön otetun, opetussuunnitelman myötä perusopetukseen tullutta ohjelmoinnin opetusta. Tämän tutkielman yhtenä tavoitteena on selvittää, kuinka opettajat ovat reagoineet tähän muutokseen. Kaikilla opettajilla on varmasti jokin kanta ohjelmoinnin opettamiseen matematiikan opetuksessa, joten koettiin mielekkääksi selvittää, minkälaisia mielipiteitä opettajilla on ohjelmointiin ja sen opetukseen liittyen.</p> <p>Tutkielmassa selvitetään, kuinka algoritmisen ja ohjelmallisen ajattelun sekä ohjelmoinnin opetusta toteutetaan sekä Suomessa että muualla maailmassa. Algoritmisen ja ohjelmallisen ajattelun termit ovat vahvasti yhteydessä niihin tavoitteisiin, mitä ohjelmoinnilla halutaan opetussuunnitelman mukaan opettaa. Tästä syystä näiden kahden termin eroja ja yhtäläisyyksiä selvennetään.</p> <p><i>Menetelmät.</i> Tutkielmaa varten toteutettiin kyselylomake, jonka tulokset kerättiin marras-joulukuussa 2018. Kysely julkaistiin matemaattisten aineiden opettajien liiton MAOL ry:n sähköpostimuodossa välittyvän "kirjeen" avulla tavoittaen näin monia yläkoulussa toimivia matematiikan opettajia. Lisäksi kyselyä jaettiin sosiaalisessa mediassa muutamissa sopiviksi katsotuissa ryhmissä.</p> <p>Kysely voidaan karkeasti jakaa kolmeen osaan. Ensimmäisessä osiossa kartoitettiin opettajien omia taustoja ja asenteita. Toisessa osassa selvitettiin opettajien resursseja, tapoja toteuttaa ohjelmoinnin opettamista sekä opettajien mielipiteitä ohjelmoinnin opettamisesta. Viimeisessä osiossa kysyttiin opettajan omaa ohjelmointitaustaa ja kouluttautumista.</p> <p>Kyselytutkimukseen osallistui 46 (19 miestä ja 27 naista) yläkoulussa matematiikkaa opettavaa opettajaa. Kyselyn tuloksia analysoitiin sekä kvalitatiivisesti että kvantitatiivisesti. Kyselylomakkeessa oli 19 kysymystä, sekä näiden jälkeen vapaa sana -osio. Kyselylomakkeessa oli sekä monivalintakysymyksiä että avoimia kysymyksiä. Kysymyksillä selvitettiin opettajien asenteita ohjelmoinnin opetusta sekä ohjelmointia kohtaan.</p> <p><i>Tulokset ja johtopäätökset.</i> Kyselyn tulosten perusteella voidaan todeta, että vain 52,2% vastaajista kokee omaavansa tarpeeksi tietokoneita ohjelmoinnin opettamiseen toteuttamiseen. Oppimateriaalien osalta vastaava luku oli 28,3% ja ajan osalta vain 13,0%. Luvut ovat huolestuttavan alhaisia ja mikäli muutosta tähän ei pian tapahdu, opettajat saattavat turhautua aiempaa enemmän nykytilanteeseen, jolloin opetuksen laatu saattaa kärsiä merkittävästi.</p> <p>Tutkielman tulokset antavat suuntaa nykytilanteesta, joten on mielekästä harkita jatkotutkimusten tekemistä. Erityisesti resurssien puuttumisten kannalta sijainti voisi antaa lisätietoa</p>			

siitä, mihin rahoitusta olisi hyvä lisätä, jotta perusopetuslain mukainen tasa-arvoinen opetus toteutuisi Suomessa.

Avainsanat – Nyckelord

Ohjelmointi, algoritmien ajattelu, CT, ohjelmoinnin opetus, opetussuunnitelma

Keywords

Programming, algorithmic thinking, CT, teaching programming, curriculum

Säilytyspaikka - Förvaringsställe - Where deposited

E-thesis

Muita tietoja - Övriga uppgifter - Additional information

Sisältö

1	Johdanto	3
2	Teoriaa ohjelmoinnin termistöstä ja opettamisesta	9
2.1	Yleisesti ohjelmoinnista	9
2.1.1	Mitä on ohjelmointi	10
2.1.2	Koodaaminen	10
2.2	Ohjelmoinnin opettamisesta (yliopistossa)	11
2.3	Algoritminen ajattelu	11
2.4	Ohjelmallinen ajattelu	14
3	Ohjelmoinnin ja ohjelmallisen ajattelun opetuksen toteutus ympäri maailmaa sekä kyselytutkimuksen käyttö	17
3.1	Ohjelmallisen ajattelun opetus Euroopassa	17
3.1.1	Alankomaat	17
3.1.2	Italia	19
3.1.3	Liettua	20
3.1.4	Ruotsi	21
3.1.5	Suomi	23
3.1.6	Yhdysvallat	23
3.2	Ohjelmoinnin opettamisesta muualla	26
3.2.1	Bulgaria	26
3.2.2	Espanja	26
3.2.3	Irlanti	26
3.2.4	Israel	27
3.2.5	Itävalta	27
3.2.6	Malta	27
3.2.7	Puola	27
3.2.8	Slovakia	28
3.2.9	Tanska	28

3.2.10	Tsekin tasavalta	29
3.2.11	Unkari	29
3.2.12	Yhdistynyt kuningaskunta (Englanti)	29
3.3	Tutkimusmenetelmänä kyselytutkimus	30
3.4	Kyselytutkimus	30
3.4.1	Ensimmäinen osio: opettajien omat taustat ja asenteet	31
3.4.2	Toinen osio: opettajien resurssit, ohjelmoinnin opettamisen toteut- taminen ja mielipiteitä ohjelmoinnin opettamisesta	33
3.4.3	Kolmas osio: opettajan oma ohjelmointitausta ja kouluttautuminen	36
3.4.4	Vapaa sana	38
3.5	Tutkimusaineisto	38
4	Tulosten analysointi	39
4.1	Kysymysten formulointi	41
4.2	Kyselylomakkeen tulokset ja huomiot	42
4.2.1	Opettajien motivaatio ohjelmoinnin opetusta kohtaan	43
4.2.2	Ohjelmoinnin opetus matematiikassa vai erikseen valinnaisena kurs- sina	47
4.2.3	Tapoja toteuttaa ohjelmoinnin opetusta yläkoulussa	48
4.2.4	Tapoja toteuttaa ohjelmoinnin opetusta matematiikan opetuksessa	48
4.2.5	Onko opettajilla tarpeeksi resursseja ohjelmoinnin opettamiseen . .	49
4.2.6	Tarvitsevatko kaikki oppilaat ohjelmoinnin opetusta	54
4.2.7	Vastaaajien huomioita ja kokemuksia	56
5	Pohdintaluku	59
5.1	Jatkotutkimusaiheita	60
6	Johtopäätösluku	63
7	Liitteet	65

Luku 1

Johdanto

Tämän Pro Gradu -tutkielman aiheena on ohjelmoinnin opetus matematiikan opetuksessa ja erityisesti yläkouluissa. Viimeisin perusopetuksen opetussuunnitelma, vuonna 2014 opetushallituksen hyväksymä, otettiin käyttöön alakoulun vuosiluokkien osalta 1.8.2016 [OPS 2014]. Yläkoulun vuosiluokkien 7 - 9 osalta uusi opetussuunnitelma otettiin porrastetusti käyttöön vastaavasti vuosina 2017, 2018 ja 2019 [OPS 2014]. Tätä asiakirjaa noudattavat kaikki Suomen peruskoulut, joten on mielekästä käydä ensin läpi, mitä opetussuunnitelmassa sanotaan ohjelmoinnin opetuksesta ja opettamisesta yläkoulussa. Otettakoon huomioon, että opetussuunnitelman osiossa, joka on kohdistettu yläkouluille, sanalla **ohjelmointi** tai sen johdannaisilla on vain kahdeksan esiintymää. Kun koko asiakirjassa niitä on 19 kappaletta. Vertailun vuoksi mainittakoon, että edellisessä, eli vuoden 2004 opetussuunnitelmassa [OPS 2004] sanaa **ohjelmointi** tai sen johdannaisia ei ollut yhtään. Uusimmassa opetussuunnitelmassa yläkoulua koskevassa osiossa ainoat esiintymät eivät rajoitu matematiikan osioon, vaan ohjelmoinnista on myös maininta *Käsityö* luvun alla. Ensimmäinen maininta ohjelmoinnista opetussuunnitelmassa yläkoulun osalta kuuluu seuraavasti:

"*Käytännön taidot ja oma tuottaminen*: Oppilaita kannustetaan oma-aloitteiseen tieto- ja viestintäteknologian hyödyntämiseen erilaisissa oppimistehtävissä sekä eri tehtäviin sopivien työtapojen ja välineiden valintaan. Heidän käsityksensä eri laitteiden, ohjelmistojen ja palveluiden käyttö- ja toimintalogiikasta syvenee. He harjaantuvat systematisoimaan, organisoimaan ja jakamaan tiedostoja sekä valmistamaan erilaisia digitaalisia tuotoksia itsenäisesti ja yhdessä. Ohjelmointia harjoitellaan osana eri oppiaineiden opintoja."

Äskeinen lainaus oli otettu opetussuunnitelman [OPS 2014] kohdasta *Laaaja-alainen osaaminen* alaluvusta *Tieto- ja viestintäteknologian osaaminen* (L5). Yksi kysymys, johon tämä tutkielma pyrkii vastaamaan on, missä määrin opetussuunnitelman kuvaus toteu-

tuu ohjelmoinnin osalta. Tutkielman fokuksessa on kuitenkin se, kuinka Suomessa matematiikkaa yläkoulussa opettavat opettajat toteuttavat ohjelmoinnin opetusta osana matematiikan opetusta. Näitä Suomessa toteutettavia tapoja verrataan muissa maissa toteutettavaan [European Schoolnet (2015)], [Mannila, L. ym. (2014)] ohjelmoinnin opettamiseen ja opetukseen. Ohjelmoinnin lisäksi avataan hieman *algoritmisen* ja *laskennallisen ajattelun* taustoja, sillä Suomessakin usein ohjelmoinnin opetuksen taustalla on juuriinkin algoritmisen ajattelu. Suomen lisäksi muita maita, joiden ohjelmoinnin ja algoritmisen ajattelun sekä Computational Thinking (CT) opetusta tässä tutkielmassa on käyty läpi, ovat mm. Italia, Liettua, Alankomaat, Ruotsi ja Yhdysvallat.

Ohjelmoinnin opetuksessa on paljon huomattavia eroja maailmanlaajuisesti, joten seuraavaksi käydään näistä erilaisista tavoista läpi vain muutamia, jotka on kuvattu artikkelissa [Mannila, L. ym. (2014)]. Italiassa on joissakin kouluissa käytössä robottilaboratoriot, joita käytetään ohjelmoinnin opetuksen osana [Mannila, L. ym. (2014)]. Liettuassa ohjelmointia opetettiin aluksi oppiaineen informatiikka nimellä (1980-luvulla) ja samoihin aikoihin isoimmissa sanomalehdissä jopa julkaistiin ohjelmointiopitunteja [Mannila, L. ym. (2014)]. Sittemmin vuonna 2005, vuosiluokkien 5 - 10 opetussuunnitelmaa uudistettiin huomattavasti ja informatiikka oppiaine muuttui *informaatioteknologiaiksi* [Mannila, L. ym. (2014)]. Samalla fokus kohdistui tietokonelukutaitoon (computer literacy), jonka seurauksena informatiikan perusteiden opetuksesta on tullut melko heikkoa [Mannila, L. ym. (2014)]. Alankomaissa vuonna 2012 laaditun raportin myötä on tehty aloite, joka uudistaisi täysin informatiikan valinnaisuudeen oppiaineena HA-VOn ja VWOn yläluokilla [Mannila, L. ym. (2014)]. Ruotsissa tietojenkäsittelytiede keskittyy oppiaineena peruskouluissa lähinnä digilukutaitoon, vaikkakaan kyseistä oppiainetta ei ole sellaisenaan pakollisissa kouluissa, vaan sisällöt tarjotaan muissa oppiaineissa kuten teknologia ja matematiikka [Mannila, L. ym. (2014)]. Ruotsissa uusi opetussuunnitelma ohjelmoinnin opetukseen on tullut ottaa käyttöön syksyyn mennessä vuonna 2018 [Nordén ym.]. Yhdysvalloissa varsinaista ohjelmointia pääsee opiskelemaan vasta lukiossa (luokat 9 - 12), mutta aiemmilla luokilla on painotettu enemmän ohjelmallisen ajattelun (Computational Thinking) opetukseen [Mannila, L. ym. (2014)]. Näistä kaikista kerrotaan lisää myöhemmin luvussa 3.

Aiemman lisäksi European Schoolnetin raportista [European Schoolnet (2015)] käy ilmi monia monia erilaisia tapoja toteuttaa ohjelmoinnin opetus opetussuunnitelmassa ja näiden lisäksi niihin liittyviä yksityiskohtia. Seuraavaksi käydään näistä muutama läpi. European Schoolnetin raportin [European Schoolnet (2015)] mukaan, Israel on ensimmäinen maa, joka on integroinut koodauksen (ohjelmoinnin) opetuksen opetussuunnitelmaan jo vuonna 1976. European Schoolnetin raportin [European Schoolnet (2015)] mukaan useassa Euroopan maassa, kuten esimerkiksi Itävallassa (Ohjelmistokehitys, Softwa-

re Development), Irlannissa (Koodaus, Coding), Maltassa (Tietokone-opinnot, Computer studies), Bulgariassa (Informatiikka, Informatics) ja Englannissa (Tietokonetoiminta, Computing) on monen muun maan lisäksi käytössä ohjelmoinnin opetus omana oppiaineenaan. Suomi on samaisen raportin [European Schoolnet (2015)] mukaan yksi maista, joissa ohjelmointia opetetaan oppiainerajat ylittävästi, eli ohjelmoinnin opetusta on sisällytetty muihin oppiaineisiin, kuten matematiikka, fysiikka, kemia, teknologia, informatiikka, mutta myös muita aineita on ja toteutukset vaihtelevat joissakin maissa alueellisesti tai koulun omasta (paikallisesta) opetussuunnitelmasta riippuen. Raportissa Suomen mainitaan kuitenkin tulevan oleva ensimmäinen maa, jossa otetaan käyttöön koodaaminen (ohjelmointi) puhtaasti oppiainerajat ylittävällä lähestymistavalla.

Yleisen tavan mukaan, myös tässä tutkielmassa on tutkinnallinen osuus. Tätä tutkielmaa varten toteutettiin kyselylomake [Kyselylomake], joka on suunnattu yläkoulun matematiikan opettajille. Kyselyn vastaukset kerättiin marras-joulukuussa 2018 sekä matemaattisten aineiden opettajien liiton MAOL ry:n sähköpostina kulkevan verkkokirjeen avulla että sosiaalisessa mediassa aineenopettajille suunnattujen ryhmien kautta. Kysely sijaitsi Helsingin yliopiston E-lomake -palvelussa, johon mahdollisille osallistujille jaettiin linkki. Kyselylomakkeeseen kuuluu 19 kysymystä, joista kymmenen on suljettuja kysymyksiä ja loput yhdeksän ovat avoimia. Kysymysten avulla pyritään kartoittamaan mm. opettajien asenteita ohjelmointia ja sen opetusta kohtaan sekä ovatko opettajat kokeneet, että opetukseen tarvittavia resursseja olisi ollut tarpeeksi. Kysely pyrkii vastaamaan kysymyksiin, kuten onko opettajilla tarpeeksi resursseja toteuttaa ohjelmoinnin opetusta sekä kuinka opettajat ovat toteuttaneet ohjelmoinnin opetusta. Vastauksia tuli yhteensä 46 kappaletta, joten otanta on valtakunnan tasolla pienehkö. Kyselyyn pyrittiin saamaan vastauksia ympäri maata kaikenlaisista yläkouluista. Kuitenkin kyselyn otanta on tarpeeksi suuri, että sitä voidaan pitää suuntaa antavana valtakunnallisesta nykytilanteesta.

Tutkielmassa tarkastellaan myös hieman termien *ohjelmointi* ja *koodaus* välisiä eroja, sillä nämä kaksi termiä menevät puhekielessä helposti sekaisin. Silloin tällöin sanomalehdistä [El Kamel, S. (2018)] voi lukea kuinka nämä termit menevät jopa kirjoitetussa kielessä sekaisin. Lyhyesti sanottuna, koodaamisella tarkoitetaan ohjelmointikäskyjen (lähdekoodin) kirjoittamista, jonka tietokoneella ohjelmointikieltä ymmärtävä ohjelmisto voi kääntää konekielelle (ykkösiksi ja nolliksi) ja suorittaa. Sen sijaan ohjelmointi on huomattavasti laajempi käsite, joka pitää sisällään aiemmin määritellyn termin koodaaminen. Ohjelmoinnille on monta määritelmää, mutta tässä tutkielmassa käytetään samaa määritelmää, kuin European Schoolnetin tuottamassa kyselyssä [European Schoolnet (2015)]. Suomeksi käännettynä määritelmä kuuluu seuraavasti:

Ohjelmointi on prosessi, jossa kehitetään ja toteutetaan joukko erilaisia tai monenlaisia käskyjä, jotka saavat tietokoneen suorittamaan tietyn tehtävän. Tietokoneen avulla

voi tällöin ratkaista pulmia tai ongelmia sekä tarjota ihmisvuorovaikutusta.
[European Schoolnet (2015)]

Yksinkertaisemmin muotoiltuna tietokoneohjelmointi on prosessi, johon kuuluu suoritettavissa olevan tietokoneohjelman suunnitteleminen ja toteuttaminen. Tämä tietokoneohjelma suorittaa tietyn tietokoneella toteutettavissa olevan tehtävän. Tämän lisäksi ohjelmointiin kuuluu tehtävät kuten analysointi, algoritmien tuottaminen, algoritmien tarkkuuden määrittäminen ja resurssien käyttö sekä algoritmien toteuttaminen valitulla ohjelmointikielellä. Ohjelmoinnin päämääränä on löytää joukko käskyjä, joilla saadaan automatisoitua annetun pulman tai ongelman ratkaisemisen suorittaminen.

Opetussuunnitelmassa ensimmäisen luvun *Paikallisen opetussuunnitelman merkitys ja laadinta* ensimmäinen alaluku *Opetussuunnitelman perusteet ja paikallinen opetussuunnitelma* alkaa seuraavin sanoin:

Perusopetuksen ohjausjärjestelmän tarkoituksena on varmistaa koulutuksen tasa-arvo ja laatu sekä luoda hyvät edellytykset oppilaiden kasvuille, kehitykselle ja oppimiselle. Ohjausjärjestelmän normiosan muodostavat perusopetuslaki ja -asetus, valtioneuvoston asetukset, opetussuunnitelman perusteet sekä paikallinen opetussuunnitelma ja siihen perustuvat lukuvuosisuunnitelmat. Järjestelmän eri osat uudistuvat, jotta opetuksen järjestämisessä pystytään ottamaan huomioon muutokset koulua ympäröivässä maailmassa ja vahvistamaan koulun tehtävää kestävän tulevaisuuden rakentamisessa.

Jotkut opettajat ovat saattaneet ihmetellä, miksi ohjelmointia tulisi ylipäättään opettaa kouluissa. Yksi European Schoolnetin raportin [European Schoolnet (2015)] perusteluista, miksi oppilaat tai opiskelijat tarvitsevat "tietokone- ja koodaustaitoja (computing and coding skills)", on pula ICT-taitoja omaavista työntekijöistä: vuoteen 2020 mennessä Eurooppa saattaa kokea ainakin 800 000 laskenta- (computing) tai informatiikka (informatics) -ammattilaisen työvoimapulan. Toinen tärkeä perussyy [European Schoolnet (2015)] on se, että koodaustaidot auttavat ymmärtämään nykypäivän digitalisoitunutta yhteiskuntaa ja vaalia sekä edistää 2000-luvun taitoja kuten ongelmanratkaisua, luovuutta, ja loogista ajattelua.

Suora lainaus uusimmasta opetussuunnitelmasta vuosiluokat 7 - 9 Laaja-alaisista opimistavoitteista Itsestä huolehtiminen ja arjen taidot (L3):

"... Eri oppiaineiden opetuksessa ja muussa koulutyössä oppilaat oppivat ymmärtämään teknologian kehitystä, monimuotoisuutta ja merkitystä omassa elämässä, kouluyhteisössä ja yhteiskunnassa. Oppilaita opetetaan ymmärtämään myös teknologian toimintaperiaatteita ja kustannusten muodostumista sekä harjoittelemaan sen vastuullista käyttöä

teknologisten ideoiden kehittämistä ja mallintamista. Oppilaiden kanssa pohditaan teknologiaan liittyviä eettisiä kysymyksiä ja tulevaisuuden mahdollisuuksia. ... "

Suora lainaus Laaja-alaisista oppimistavoitteista Tieto- ja viestintäteknologinen osaaminen (L5):

"Matematiikan opetuksen tavoitteet vuosiluokilla 7. - 9. Opetuksen tavoitteista kyseessä on Työskentelyn taidot T9, joka kuuluu seuraavasti: *opastaa oppilasta soveltamaan tietojen ja viestintäteknologiaa matematiikan opiskelussa sekä ongelmien ratkaisemisessa.* Tähän tavoitteisiin liittyvät sisältöalueet S1 - S6 ja laaja-alaisesta osaamisesta L5. Sekä toinen opetuksen tavoite T20, joka kuuluu *ohjata oppilasta kehittämään algoritmista ajatteluun sekä taitojaan soveltaa matematiikkaa ja ohjelmointia ongelmien ratkaisemiseen.* Tähän tavoitteeseen liittyy sisältöalue S1, sekä laaja-alaiset osaamisalueet L1, L4, L5 ja L6."

Sisältöalue "S1 Ajattelun taidot ja menetelmät" kohdassa lukee "Sovelletaan itse tehtyjä tai valmiita tietokoneohjelmia osana matematiikan opiskelua."

Matematiikan päättöarvioinnin kriteerit hyvälle osaamiselle (arvosanalle 8) oppimäärän päättyessä arvosanan kahdeksan osaamiseen vaaditaan, että "Oppilas osaa soveltaa algoritmisen ajattelun periaatteita ja osaa ohjelmoida yksinkertaisia ohjelmia."

Tässä luvussa esitellään tutkittavana oleva aihe, joka on ohjelmoinnin opetus matematiikan opetuksessa. Matematiikka on tieteenä ollut olemassa jo vuosituhansia ja oppiaineenakin sitä on opetettu muodossa tai toisessa niin pitkään kuin koulujärjestelmä on ollut. Ohjelmointi sen sijaan on melko uusi ilmiö. Vaikkakin ensimmäinen muistiin tallennetun ohjelman toimintaperiaate keksittiin ja julkaistiin First Draft of a Report on the EDVAC -raportissa vuonna 1945, niin ensimmäisten nykyaikaisten tietokoneiden valmistuminen tapahtui vasta 1949 [Orponen, P.]. Ohjelmoinnin opettamista Suomessa sen sijaan on alettu toteuttaa vasta virallisemmin viimeisimmän opetussuunnitelman myötä.

Muita kysymyksiä joihin tämä tutkielma pyrkii löytämään vastauksen, on: Onko kyselyyn vastaajien työkokemuksen (kysymys 2) määrällä yhteyttä tai vaikutusta vastaajan asenteeseen ohjelmointia (kysymys 3) ja sen opetusta (kysymys 4) kohtaan. Ovatko opettajat opettaneet mielellään ohjelmointia vai eivät, entä onko opettajan omilla taidoilla ollut merkitystä tässä yhteydessä. Oma asenne ja osaaminen, miten nämä kaksi korreloivat keskenään. Kysymykset 17, 18 ja 19 käsittelevät opettajan osaamista ja mielenkiintoa. Kysymykset 3 ja 4 käsittelevät vastaajan asennetta. Mielenkiintoinen kysymys itsessään on se, että onko opettajalla ollut tarpeeksi resursseja (tietokoneita, oppimateriaaleja ja aikaa). Ovatko opettajat olleet tyytyväisiä ohjelmoinnin opetukseen koulussaan (kysymys 11). Ovatko materiaalit olleet hyviä (kysymys 14), yhdistettynä onko opiskellut alaa (ky-

symys 17 ja 18). Onko ohjelmoinnin opetusta toteutettu osana matematiikan opetusta, vai täysin erikseen (kysymys 16).

Luku 2

Teoriaa ohjelmoinnin termistöstä ja opettamisesta

Tässä luvussa käsitellään ohjelmoinnin opetukseen ja oppimiseen liittyvää teoriaa sekä aiheeseen liittyviä termejä. Aluksi selvennetään mitä ohjelmoinnilla tarkoitetaan yleisesti sekä erityisesti tämän tutkielman yhteydessä. Toiseksi annetaan määritelmä *algoritmiselle ajattelulle*. Seuraavaksi käydään läpi, mitä tarkoitetaan termillä *computational thinking* ja kuvaillaan, miten eri maissa tämä on toteutettu. Tämän jälkeen käsitellään teoriaa liittyen ohjelmoinnin opettamiseen ja miten tämä teoria näyttäytyy tämän tutkielman valossa. Lopuksi käydään läpi edellisten määritelmien yhtymäkohdat nykypäivän ohjelmoinnin opetukseen yläkoulussa. Mainittakoon, että tässä tutkielmassa ei haluttu tarkoituksella antaa merkittävää painoarvoa ohjelmoinnin opetukselle matematiikan opetuksessa lukiossa, sillä keskeisin teos, joka tähän tutkielmaan vaikutti alkuvaiheessa oli perusopetuksen opetussuunnitelma.

2.1 Yleisesti ohjelmoinnista

Ohjelmoinnista kuulee puhuttavan joskus koodaamisena. Tämä termi koetaan kuitenkin alatyylisenä ja vähättelevänä terminä ohjelmoinnista, sillä koodauksesta puhuttaessa puhutaan lähinnä vain lähdekoodin kirjoittamisesta [Mannila, L. ym. (2014)]. Ohjelmointi on kuitenkin paljon laajempi käsite kuin vain lähdekoodin kirjoittamista tai sen ymmärtämistä. Tässä luvussa tarkennetaan, mitä ohjelmoinnilla tarkoitetaan. Lyhyesti voidaan todeta ja sanoa, että ohjelmointi sisältää koodaamisen, eli lähdekoodin kirjoittamisen. Ohjelmointi pitää sisällään kuitenkin paljon muutakin.

2.1.1 Mitä on ohjelmointi

Määritelmä ohjelmoinnille annettiin johdannossa, joten sitä ei toisteta tässä uudestaan.

Ohjelmointiprosessi etenee seuraavasti: Aluksi valitaan pulma, joka halutaan ratkaista. Pulma jaetaan osiin tai pienempiin pulmiin, jotka ymmärretään selkeästi. Seuraavaksi jokaiseen näistä pienemmistä pulmista kirjoitetaan tietokoneohjelma (koodi), joka ratkaisee/hoitaa yhden pienen pulman. Näitä ohjelmia yhdistämällä saadaan ratkaistua alkupe-
räinen isompi pulma. Yleinen ohjelmointiprosessi kuvattiin äsken, ja aiemmin mainittujen vaiheiden lisäksi voi ohjelmoija joutua jakamaan useasti pulman useaan pienempään pulmaan, ja mahdollisesti vielä jo jaettu pienempi pulma aiempaa pienemmäksi. Vasta kun pulman ratkaisu on kirjoitettavissa tietokoneohjelmaksi ja koodiksi, voi ohjelmoija edetä ohjelmiston rakentamisessa. Oikean ohjelmiston rakentaminen vaatii luonnollisesti vielä huomattavan määrän laajaa ymmärrystä eri osien välisistä yhteyksistä, kuten miten eriyttää käyttöliittymä ja sovelluslogiikka. Lisäksi täytyy ottaa huomioon tietoturvan vaikutus sekä monet, monet muut näkökulmat mitä tulee ohjelmiston kehitykseen. Oikean ohjelmiston tuottamista yläkoulussa ei kuitenkaan harjoitella, mutta oppilaiden on hyvä hahmottaa miten pienempien ongelmien ratkaiseminen on osa isomman ongelman ratkaisemista.

2.1.2 Koodaaminen

Koodaamiselle ei anneta yleensä kirjallisuudessa formaalia määritelmää, mutta usein koodaamisesta puhuttaessa tarkoitetaan vain lähdekoodin kirjoittamista. Aiheeseen perehtymättömälle ohjelmointi lieneekin näyttäytyvän pelkkänä lähdekoodin kirjoittamisena. Jos ohjelmointi nähdään vain rutiinitehtävänä antaa ohjelmointikäskyjä tietokoneelle, jotta saadaan ratkaistua joukko tavanomaisia ongelmia, ei tämä riitä tärkeiden taitojen, kuten CT:n, opetuksessa [Mannila, L. ym. (2014)]. Termille *Computational Thinking* (CT) ei valitettavasti ole suomenkielestä löytynyt vielä vakiintunutta käännöstä, mutta eräs näistä käännöksistä on *ohjelmallinen ajattelu*. Tämä termi määritellään tarkemmin myöhemässä luvussa. Koodaaminen on vain murto-osa ohjelmointiprosessia ja se voidaan nähdä viimeisenä vaiheena toteutettaessa ratkaisua, johon on päädytty muiden vaiheiden kautta kuten analysointi, osiin jakaminen, suunnittelu [Mannila, L. ym. (2014)]. Viimeisimpänä mainitut vaiheet ovat tärkeitä osia CT:tä [Mannila, L. ym. (2014)]. Tästä syystä olisi tärkeää välttää termiä "koodaaminen" ohjelmoinnista puhuttaessa. Tämä olisi karrikoiden sama asia, kuin puhuttaisiin kynän liikuttamisesta tai viivojen piirtämisestä muotokuvaa tehdessä. Kynän liikuttaminen on vain murto-osa taideteosta tehdessä.

2.2 Ohjelmoinnin opettamisesta (yliopistossa)

Ohjelmointia on viimeisen kuudenkymmenen vuoden ajan opetettu monin eri tavoin. Osa näistä opetustavoista on jäänyt historiaan ja osan tilalle on tullut aivan uusia tapoja. Vuosien saatossa on nähty ohjelmoinnin opetuksessa monia muutoksia kuten ensimmäisen ohjelmointikurssin opettamisesta käyttäen symbolista konekieltä (assembly language) tai Fortran I:ia, on siirrytty opettamaan sivistyneitä funktionaalisia- ja olio-ohjelmointikieliä [Bennedsen, J., ym. (2008)]. Saman ajan aikana on tapahtunut valtavasti muutoksia tietotekniikkarintamalla. Tänä aikana on siirrytty tietokoneista, joita käyttävät ainoastaan ammatti-operaattorit tietokoneisiin, joita pienet lapset käyttävät pelamiseen [Bennedsen, J., ym. (2008)]. Tänä aikana on tapahtunut myös siirtymä reikäkorttien käyttämisestä syötteenä (input) ja odottamalla tunteja tulostetta (output) tietokoneen ohjelman suorittamisen seurauksena, siihen, että painaessa tietokoneen näppämistön näppäintä konekäsky käännetään ja suoritetaan välittömästi [Bennedsen, J., ym. (2008)].

Tapoja opettaa ohjelmointia on nykypäivänäkin useita: kisällioppiminen integroituna luentoihin ja tehtäviin; ohjelmoinnin opettamista ongelmalähtöisellä (Problem Based Learning) metodilla; verkosta löytyvien opetusvideoiden ja verkkokurssien käyttäminen IT-alan johdatuskursseilla.

2.3 Algoritminen ajattelu

Kuten Gerald Futschek huomauttaa [Futschek, G.(2006)], on algoritmille monta määritelmää kirjallisuudessa. Futschek antaa kuitenkin seuraavan määritelmän:

Algoritmi on metodi tai menetelmä, jolla voidaan ratkaista ongelma, joka koostuu tarkoin määritellyistä ohjeista/käskyistä.

Tässä tutkielmassa käytetään samaa määritelmää kuin Gerald Futschek [Futschek, G.(2006)]. Algoritmia voi ajatella helpommin suomenkielen sanan *resepti* avulla. Aivan kuten algoritmissa, myös reseptissä on selkeät vaiheet, jotka kuvaillaan mahdollisimman yksityiskohtaisesti. Algoritmin ja reseptin merkittävin ero lienee se, että reseptiä lukee ihminen ja algoritmia tulkitsee tietokone. Toisena kohtalaisen merkittävänä erona voidaan mainita kieli millä resepti on kirjoitettu. Mikäli esimerkiksi juustokakun ohje on kirjoitettu venäjäksi, ei välttämättä suomalainen mestarikokkikaan pysty seuraamaan tätä reseptiä. Sama pätee algoritmeissa, ne on kirjoitettu aina tietyllä ohjelmointikielellä. Mikäli tietokone yrittää suorittaa algoritmia, joka on kirjoitettu esimerkiksi Javalla ja yrittää tulkita sitä kuin se olisi kirjoitettu Pythonilla, ei tämä toimi.

Nyt kun määritelmä algoritmille on käyty läpi, voidaan selventää, mitä tarkoitetaan

algoritmisella ajattelulla *algorithmic thinking*. Ennen kuin Futschek kuvailee taitoja, jotka liittyvät algoritmien luomiseen ja ymmärtämiseen, hän mainitsee algoritmisen ajattelun olevan yksi tärkeimmistä kompetensseista (pystyvyyksistä ja taidoista), joka voidaan saavuttaa opiskelemalla informatiikkaa (Informatics). Tämän jälkeen Futschek luettelee tarkentaen aiemmin mainitut taidot:

- kyky analysoida annettuja pulmia.
- kyky täsmällistää ja tarkentaa pulma tai ongelma tarkasti.
- kyky löytää riittävät perustoiminnot tai perustoimet annettuun pulmaan, joilla saadaan pulma tai ongelma ratkaistua.
- kyky muodostaa oikea algoritmi annettuun pulmaan tai ongelmaan käyttäen perustoimintoja tai perustoimia.
- kyky ajatella kaikkia mahdollisia erikois- ja normaalitapauksia pulmasta tai ongelmasta.
- kyky parantaa algoritmin tehokkuutta.

Tässä kohtaa on paikallaan tarkennus, mitä tarkoitetaan perustoimilla tai perustoiminnoilla. Ohjelmoinnissa näillä perustoimilla tai perustoiminnoilla tarkoitetaan yhtä perustavanlaatuisesta toimintoa, joka ohjelmointikielessä on. Esimerkiksi Java ohjelmointikielessä voitaisiin määritellä kokonaislukumuuttuja (int, englanninkielen sanasta integer, suom. kokonaisluku), jota nimitetään nimellä "luku" ja jonka arvo on viisi, seuraavalla tavalla:

```
int luku = 5
```

Tämä olisi esimerkki perustavanlaatuisesta toimesta tai toiminnosta. Toisena esimerkkinä voitaisiin käydä läpi, miten Java ohjelmointikielellä tulostettaisiin merkkijono (String-tyyppinen muuttuja) **hyvää huomenta**. Tämä tapahtuisi seuraavasti:

```
String merkkijono = "hyvää huomenta"
```

Mikäli aiemman sijaan äsken määritelty merkkijono haluttaisiin tulostaa näytölle, onnistuisi tämä seuraavasti:

```
System.out.print(merkkijono)
```

Tällöin näytölle tulostuisi merkkijono "hyvää huomenta". Kaikki äsken mainitut koodinpätkät olivat esimerkkejä perustavanlaatuisista toimista, joihin Futschek viittaa luetellessaan taitoja, joita tarvitaan algoritmisessa ajattelussa. Nämä perustavanlaatuiset toimet tekevät siis ainoastaan yhden yksittäisen toimen, joita ei toisin sanoen voida (tai ole järkevää) jakaa enää pienempiin osiin. Aiempaa analogiaa resepteihin voidaan käyttää tässäkin yhteydessä, jotta saadaan selkeämpi käsitys siitä, mitä *toimella* tarkoitetaan. Reseptissä yksikäsitteinen perustavanlaatuinen toimi voisi olla esimerkiksi "Sekoita kuivat aineet keskenään". Kyseisessä reseptiesimerkissä kokin tulee tietää mitkä ovat kyseessä olevat kuivat aineet sekä mitä tarkoittaa sekoittaminen tai milloin seos on sekoitettu. Vastaavasti ohjelmoinnissa ohjelmoijan tulee tietää mitä mikäkin käsky tekee.

Kuvattuaan taidot, joita tarvitaan algoritmien ymmärtämiseen ja luomiseen, siis algoritmiseen ajatteluun, Futschek kertoo, että hänelle algoritmisessa ajattelussa on vahva luomisaspekti. Tätä luomisaspektia hän kuvailee seuraavasti: uusien algoritmien luominen, jotka ratkaisevat annetut pulmat tai ongelmat. Mikäli joku haluaa tehdä näin, hän tarvitsee kykyä, jota kutsutaan algoritmiseksi ajatteluksi.

Nyt kun on saatu tarpeeksi kattava ymmärrys siitä, mitä algoritmisella ajattelulla tarkoitetaan, voidaan seuraavaksi käydä läpi, mitä Futschek sanoo algoritmisen ajattelun opettamisesta. Ensiksi hän myöntää, että kysymykseen "Kuinka opettaa algoritmista ajattelua?" on yhtä vaikeata vastata kuin kysymykseen "Kuinka opettaa luovuutta?". Käytännöllisenä vastauksena hän pitää monen ongelman tai pulman ratkaisemista. Erityisesti aloittelijoille nämä pulmat tai ongelmat tulee valita erityisellä huolella. Lisäksi hän ottaa kantaa valintakysymykseen kannattamalla korkean tason kieltä sekä ongelmalähtöistä lähestymistapaa algoritmin muotoiluun. Futschekin mielestä pseudokoodi täyttää nämä kriteerit. Pseudokoodilla tarkoitetaan oikean ohjelmointikielen näköistä kieltä, joka on kirjoitettu kuvaamaan tiettyä koodinpätkää. Pseudokoodi ei kuitenkaan ole mitään oikeaa ohjelmointikieltä. Oppimisen tukena Futschekin mielestä toimii erittäin hyvin algoritmien visualisointi. Havainnollistetaan seuraavaksi, mitä tarkoitetaan pseudokoodilla ja algoritmien visualisoinnilla.

Esimerkki pseudokoodista:

Toista seuraavaa kunnes lukujono on suuruusjärjestyksessä.

Jos kahdesta peräkkäisestä käsittelyssä olevasta luvusta vasemmanpuoleinen on suurempi, niin vaihda näiden lukujen paikkaa jonossa. Muussa tapauksessa älä tee mitään.

Ota kaksi seuraavaa lukua käsittelyyn, joista ensimmäinen luku on kahden aiemman jälkimmäinen luku.

Jos kahden peräkkäisen luvun jälkeen ei ole seuraavia lukuja, siirry lukujonon alkuun.

Tai lyhyemmin käyttäen symboleita (sekä vähäistä määrää tekstiä):


```

if(numbers in order)
    stop
else
    for  $i = 0$  to arrayLength
        if( $x_i > x_{i+1}$ )
            swap  $x_i$  and  $x_{i+1}$ 
        if  $x_{i+1}$  not exist
             $i = 0$ 

```

Esimerkki Algoritmin visualisoinnista. Hakasulkeilla kuvataan tässä taulukon sisältämiä lukuja. Alleviivatut luvut ovat nyt käsittelyssä olevia lukuja, eli lukuja joiden suuruuksia verrataan toisiinsa.

Alkutilanne $[3, 2, 5, 6, 4]$

1. vaihe	$[\underline{3}, \underline{2}, 5, 6, 4]$
2. vaihe	$[2, \underline{3}, \underline{5}, 6, 4]$
3. vaihe	$[2, 3, \underline{5}, \underline{6}, 4]$
4. vaihe	$[2, 3, 5, \underline{6}, \underline{4}]$

5. vaihe	$[\underline{2}, \underline{3}, 5, 4, 6]$
6. vaihe	$[2, \underline{3}, \underline{5}, 4, 6]$
7. vaihe	$[2, 3, \underline{5}, \underline{4}, 6]$
8. vaihe	$[2, 3, 4, \underline{5}, \underline{6}]$
Lopputilanne	$[2, 3, 4, 5, 6]$

Äskeinen visualisointi oltaisiin voitu toteuttaa myös esimerkiksi erivärisillä palikoilla, joissa numero lukisi palikan kyljessä. Myös luovempikin visualisointi olisi toki voinut soveltua, kuten esimerkiksi internetin videopalvelusta (YouTube) löytyvällä videolla oli tehty. Videolla oli rivissä järjestettäviä lukuja vastaava määrä ihmisiä, joiden käsissä oli lappu, jossa luki numero. Rivissä olevien ihmisten piti päästä lukuja vastaavaan suuruusjärjestykseen, mutta mieleenjäävän videosta teki se, että ihmiset liikkuivat kansanmusiikin tahtiin tanssien.

2.4 Ohjelmallinen ajattelu

Äskeisessä luvussa 2.3 käsiteltiin algoritmista ajattelua, tässä luvussa käsitellään toista termiä "Computational thinking" (lyhenteenä CT), kuten myös lähteessä

[Mannila, L. ym. (2014)]. Tässä tutkielmassa käytetään jatkossa lyhennettä CT tarkoittamaan tätä ohjelmallista ajattelua. Mannilan ja muiden mukaan useita määritelmiä termille CT on ollut jo vuodesta 1996 lähtien, jolloin Simon Papert mainitsi siitä matematiikan opetusta koskevassa artikkelissa. Kirjoittajien [Mannila, L. ym. (2014)] mukaan Papert mainitsee tietokoneiden käytön pulmien ja ongelmien ratkaisussa tavoilla, jotka luovat ideoita (*forge ideas*) ja antaa ihmisten analysoida paremmin ja selittää ongelmia tai pulmia, ratkaisuja ja niiden välisiä yhteyksiä. Tutkielmassa [Mannila, L. ym. (2014)] mainitaan tähän läheisesti liittyvän käsitteen *computational literacy*, eli laskennallisen (ohjelmointikielten ymmärtämiseen tarvittavan) lukutaidon olemassaolosta, jonka on luonut Vee vuonna 2013. Vee määrittelee [Vee, A. (2013)] laskennallisen lukutaidon seuraavasti:

Laskennallinen lukutaito on kokoelma kykyjä ja taitoja jakaa kompleksi tai monimutkainen prosessi pieniin prosedureihin ja sitten ilmaista - tai "kirjoittaa" nämä proseduurit käyttäen teknologiaa, kuten koodia, siten, että jokin entiteetti tai olio (muu kuin ihminen) voi "lukea" tai "ymmärtää" sitä, kuten esimerkiksi tietokone.

Kuitenkin myöhemmän määritelmän CT:lle antoi Jeannette Wing [Mannila, L. ym. (2014)] vuonna 2006, joka tuli siis kymmenen vuotta Papertin jälkeen. Wingin ehdottama määritelmä on esitetty suomennettuna alla:

Universaalisti sovellettavissa oleva joukko asenteita, kykyjä ja taitoja, joita haluaisivat oppia ja käyttää innokkaasti kaikki, eivätkä vain tietojenkäsittelytieteilijät.

Myöhemmin Wing antoi konkreettisemmän määritelmän todeten, että CT voidaan ymmärtää seuraavasti:

Ajatteluprosessit, jotka liittyvät ongelmien ja pulmien muotoiluun sekä niiden ratkaisuihin siten, että ratkaisut esitetään muodossa, jossa ne voidaan suorittaa informaatiota tai tietoa käsittelevän ja prosessoivan toimijan avulla.

Näillä kahdella (CT:llä ja algoritmisella ajattelulla) määritelmällä on yhtäläisyyksiä ja samankaltaisuuksia, mutta yksimielisyyttä ei ole aikojen saatossa saavutettu – ainakaan vielä toistaiseksi. Seuraavaan listaan on kerätty aiempien määritelmien yhteispiirteitä:

- ongelman täsmällistäminen,
- taitojen hallitseminen ongelman ratkaisemiseen,
- ongelman ratkaisun siirtäminen välineelle, joka lopulta suorittaa ja tuottaa ratkaisun.

Yhtenä erona määritelmien välillä voitaneen mainita olevan se, että algoritmisessa ajattelussa tarkennetaan ongelman ratkaisemisen vaiheita ja samalla jo olemassa olevien vaiheiden parantamista. Aiemman sijaan CT:ssä kuvataan ajatteluprosessia hyvin yleisellä tasolla.

Luku 3

Ohjelmoinnin ja ohjelmallisen ajattelun opetuksen toteutus ympäri maailmaa sekä kyselytutkimuksen käyttö

3.1 Ohjelmallisen ajattelun opetus Euroopassa

Seuraavaksi käydään läpi millä tavoin Suomessa ja muissa Euroopan maissa on toteutettu ohjelmallisen ajattelun opetuksen periaatteita. Yksi merkittävä ero Suomen ja monen muun maan välillä on se, että Suomi oli ensimmäinen maa, joka otti käyttöön koodaamisen (ohjelmoimisen) puhtaasti oppiainerajat ylittävällä lähestymistavalla [European Schoolnet (2015)]. Muita maita, joiden toteutuksia käydään tässä tutkielmassa läpi, ovat Alankomaat, Italia, Liettua, Ruotsi ja Yhdysvallat.

3.1.1 Alankomaat

Käydään läpi, miten CT:n opetusta on toteutettu Alankomaissa [Mannila, L. ym. (2014)].

Alankomaissa opetuksen tavoitteet on kuvailtu melko yleisellä tasolla ja koulujen harjoitustyöskentelyyn luotetaan itsenäisen toteuttamisen puolesta luokkatilassa. Usein on olemassa ohjeita, jotka sisältävät yksityiskohtaisia tulkintoja tavoitteista, mutta näiden tulkintojen totteleminen ei ole pakollista. Käytännössä kirjojen julkaisijat tulkitsevat ydintavoitteita ja julkaisevat oppikirjoja, joita käytetään perustana itse opetusaktiviteeteille.

Oppimistavoitteet esi- ja perusopetukselle (primary education) (4 - 12 -vuotiaat) on koottu 58 yleiseen ydintavoitteeseen, joissa kuvaillaan tavoitteet hollannin, friisin ja englannin kielille; aritmetiikalle/matematiikalle; maailman ja omalle suuntautumiselle; taiteelle ja liikunnan opetukselle. Vain kymmenen näistä tavoitteista sisältävät CT:n näkökulmia, esimerkiksi seuraavasti:

Hollannin kieli:

4. Oppilaat oppivat löytämään tietoa informatiivisista ja opettavaisista (instructive) teksteistä, joihin kuuluu rakenteita, taulukoita ja digitaalisia resursseja.
6. Oppilaat oppivat järjestelemään tietoa ja mielipiteitä lukiessaan koulussa ja opiskelutekstejä ja muita opettavaisia tekstejä, ja lukiessaan muita systemaattisesti järjestettyjä lähteitä, mukaan lukien digitaaliset lähteet.
24. Oppilaat oppivat ratkomaan käytännöllisiä ja matemaattisia ongelmia tai pulmia ja esittämään selvästi päättelyään.

Vuonna 2012 alankomaiden kuninkaallinen taiteiden ja tieteiden akatemia (Koninklijke Nederlandse Akademie Van Wetenschappen) julkaisi raportin *Digitaalinen lukutaito toisen asteen koulutuksessa - Taidot ja asenteet 2000-luvulle*. Tässä raportissa on tehty useita suosituksia, kuten:

- Ottaa käyttöön uusi pakollinen oppiaine tiedotus ja kommunikointi HAVOn ja VWOn ensimmäisille vuosille. Tämän kuuluisi olla laaja ja tiivis perehdytys oppiaineeseen, kattaen merkittävimmät digitaalisen lukutaidon osa-alueet.
- Rohkaista näiden ja muiden kouluaineiden välistä vuorovaikutusta.
- Asettaa tärkeysjärjestykseen kouluttaa uusi sukupolvi opettajia, joilla on uusia kykyjä ja asenteita.

HAVO on lyhenne sanoista *Hoger algemeen voortgezet onderwijs*, joka kääntyy suomeksi korkeampi yleinen jatkokoulutus tai yleinen korkeampi jatkokoulutus. HAVO vastaa Suomen yläkoulua ja lukion yleislinjan yhdistelmää (oppilaat ovat 12 - 17 vuotiaita). VWO on lyhenne sanoista *Voorbereidend wetenschappelijk onderwijs*, joka kääntyy suomeksi valmistava tieteellinen koulutus. Tämä koulutus on kuusivuotinen, ja vastaa myös Suomen yläkoulua ja lukiota tietyin painotuksin eri tieteisiin.

Lisäksi aiemmin mainitussa raportissa CT:tä pidetään olennaisena osana digitaalista lukutaitoa. Raportin suositukset johtivat kahteen aloitteeseen:

- 1) uudistaa täysin informatiikan valinnaisuus oppiaineena HAVOn ja VWOn yläluokilla,
- 2) tutkia mitä tarkoitetaan digitaalisen lukutaidon aloitteella, mitä näkökulmia siitä halutaan luokilla 7 - 9 ja miten se sijoittuu opinto-ohjelmaan.

Vastatakseen näihin vaatimuksiin Alankomaiden opetussuunnitelmien kehityksestä vastaava instituutti teki tutkimusta, jossa selvitettiin kuinka paljon 2000 - luvun taitoja ja digilukutaitoa, mukaan lukien CT, K - 9 opetuksessa oppimistavoitteissa ja opetusmateriaaleissa ja opetuskäytännöissä. Ensimmäiset löydökset osoittivat, että digilukutaitoon on panostettu vain vähäisellä huomiolla oppimistavoitteissa, joita kuvaillaan melko yleisin termein. Opetusmateriaalit K - 6 opetuksessa hädin tuskin sisältävät näiden tavoitteiden opetusta. Vuosiluokilla 7 - 9 sen sijaan näihin tavoitteisiin kiinnitetään enemmän huomiota.

3.1.2 Italia

Lähteessä [Mannila, L. ym. (2014)] kerrotaan, miten CT:tä opetuksessa on toteutettu Italiassa kouluissa.

Vuosien 2007 ja 2012 välissä Italian koulutusjärjestelmä kävi läpi laajan reformiprosessin [Mannila, L. ym. (2014)]. Reformin tavoitteena oli muuttaa sekä koulutuksellinen lähestyminen että opetussuunnitelma organisaatio. Nykyisin Italiassa pakollisen oppimäärän kesto on kymmenen vuotta. Italiassa esiopetuksesta kahdeksannelle luokalle asti (k - 8) tietojenkäsittelytieteen ja digitaalisten teknologioiden opetus ei ole minkään tietyn aiheen vastuulla. Kansallisen opetussuunnitelman suositusten mukaan tällaiset sisällöt kuuluvat kahteen melko laajaan alueeseen:

- oppiainerajat ylittävä kansalaisen digitaalinen avaintaito, jonka yhteydet tietojenkäsittelytieteen kanssa ovat melko heikot (taito ja kriittinen asenne tieto- ja viestintätekniikan käyttämisessä työssä, elämässä ja kommunikoinnissa; tietokoneen käyttö informaation hakemiseen, arviointiin, säilyttämiseen, tuottamiseen, esittämiseen ja jakamiseen, kuten myös yhteistyön tekeminen internetin välityksellä).
- yleisen teknologian aihe-alue, joka sisältää yleisimpien tieto- ja viestintätekniikan työkalujen käytön ja "jos mahdollista" jonkin verran tietokoneen ja/tai robottien ohjelmointia (taito ja kriittinen asenne psykologista, sosiaalista ja kulttuurillista tieto- ja viestintätekniikan vaikutuksia kohtaan ja jos mahdollista, johdatus ohjelmointiin yksinkertaisilla kielillä, joilla voidaan luoda ja kehittää projekteja).

Viittaus ohjelmointiin ja robotteihin on yritys ottaa huomioon jotkut (näennäisesti) onnistuneet kokemukset, joita ovat mainostaneet innostuneet, itse-motivoituneet (ja itse-oppineet) opettajat.

Oppiainerajat ylittävä digitaalinen pystyvyys aihe-alue kattaa koko pakollisen opetuksen ajanjakson. Tämän jälkeen toisen asteen joko ensimmäisenä vuotena, tai kahtena ensimmäisenä vuotena opetetaan muusta opetuksesta erillään informatiikkaa tai tieto- ja

viestintäteknikkaan liittyvää ainetta pätevän opettajan toimesta. Tämä järjestely sisältyy vain joidenkin koulujen opetussuunnitelmaan. Tavallisen käytön ja tavallisten käsitysten yleisimmistä alustoista ja informaatioteknologian työkaluista, joita odotetaan kaikkien osaavan, lisäksi sekä soveltavien ja teknillisten tieteenalojen opiskelijoiden odotetaan oppivan jotain algoritmeista ja tietokoneohjelmoinnista.

Vaikkakin tilanteen kerrotaan voivan vaihdella hyvinkin paljon koulusta kouluun riippuen siitä, miten itsenäiset aloitteet, joita reformin tuoma autonomia edesauttaa, vaikuttavat. Erityisesti motivoituneiden opettajien työpanos ja luovuus voi johtaa luokissa mielenkiintoisiin aktiviteetteihin. Mainittavia esimerkkejä tässä yhteydessä olivat johdanto oppilas-ystävällisiin ohjelmointiympäristöihin kuten Scratch ja opetuksellisten robottilaboratorioiden kehittyminen.

3.1.3 Liettua

Esitellään seuraavaksi, millä tavalla CT:n opetus on toteutettu kouluissa Liettuassa [Mannila, L. ym. (2014)].

Liettuassa tietojenkäsittelytieteestä puhutaan termillä informatiikka (informatika). Tämän oppiaineen opetus alkoi 1980-luvun alussa ohjelmoinnilla ja koneiden koodaamisena. Ohjelmoinnin opetuksen tavoite on ongelmanratkaisukyvyyn siirtäminen ja se, että ohjelmointi on paras tapa kommunikoida koneen kanssa. Vuosina 1985 - 1986 informatiikka julistettiin pakolliseksi oppiaineeksi lukioissa Neuvostoliitossa, mukaan lukien Liettua. Tuohon aikaan informatiikan opetus oli kohdistettu algoritmien, ajattelutaitojen, abstraktion kehittämiseen ja tehtävien ratkaisemisen automatisointiin. Nämä kaikki taidot luetaan kuuluvaksi CT:hen. Oppilaille kehitettiin sadoittain mielenkiintoisia ja houkuttelevia tehtäviä. Valikoima haastavia tehtäviä "A hundred programming problems" julkaistiin Vilnassa vuonna 1986 ja Moskovassa 1993. Tieto- ja viestintäteknikan roolin ja oppilaiden sekä kouluyhteisöjen tarpeiden muuttuessa, luokkien 5 - 10 kaikkien oppiaineiden opetussuunnitelma koki merkittäviä muutoksia vuonna 2005, jolloin "informatiikka" muutettiin "informaatioteknologiaksi". Vuodesta 2005 lähtien keskiössä Liettualaisissa kouluissa on ollut "tietokonelukutaidon" (computer literacy) kehittäminen. Seurauksena informatiikan perusteiden opetuksesta on tullut melko heikkoa. Oppilaat tutustuvat informatiikan perusteisiin luokilla 5 tai 6, kun ensimmäiset IT oppitunnit pidetään, mistä osa tulisi keskittää Logon tai Scratchin opetukseen. Kuitenkin hyvin vähän oppitunteja on käytössä tähän, jolloin opetusprosessi riippuu hyvin vahvasti opettajien omista tiedoista ja aktiivisuudesta.

Luokilla 9 - 10 on tarjolla valinnainen opintomoduuli algoritmien suunnittelusta ja laatisemisesta (ohjelmoinnista), jota suositellaan seuraavilla aiheilla: käsitys algorimeista, tapoja kirjoittaa; ohjelmointikielet, kääntäjät; algoritmien valmistelu, koodaaminen ja ohjelman

suorittaminen; dialogi käyttäjän ja tietokoneohjelman välillä; tietojen syöttäminen ja tulostaminen, tulostusformaatit; algoritmien päätoimenpiteet: sijoitus, silmukka; yksinkertaiset tietotyypit; ohjelmiston kehitysvaiheet; tiedonhallinta ja oikeellisuus ohjelmassa; ohjelmointityyli ja -kulttuuri; yksinkertaisimmat algoritmit ja niiden toteutus.

Informatiikka luokilla 9 - 10 esittelee oppiaineen perusteet ja toimii lähtökohtana informatiikan opetukseen lukioon meneville sekä perehdytysjaksona niille, jotka saattavat olla kiinnostuneita menemään lukioon, joka tarjoaa tietojenkäsittelytieteen aiheisiin erikoistuneita kursseja, kuten algoritmit, tietoverkot, tai tietokannat. Informaatioteknologiaa oppiaineena luokilla 5 - 10 opettaa opettajat, joilla on monenlaista koulutusta. Joillakin on tietojenkäsittelytieteen tai matematiikan tutkinto yhdistettynä opettamiseen, kun muut ovat muiden oppiaineiden opettajia ja joilla on vähän kokemusta informaatioteknologiasta.

3.1.4 Ruotsi

Tässä alaluvussa luodaan katsaus Ruotsin CT:n opetukseen. Ruotsissa tietojenkäsittelytieteen opetuksen nykymalli perustuu 1970 - lukuun, tuolloin nimeltään informatiikka, mutta myöhemmin muutettiin informaatioteknologiaksi. Oppiaine oli olemassa valinnaisena aineena ammattikouluissa ja siitä tuli myöhemmin pakollinen toisen asteen luonnontieteellisten koulutusohjelmien opiskelijoille. Muutaman vuoden ajan opetusta tarjottiin alaja yläkouluissa, mutta myöhemmin opetus lopetettiin, sillä peruskoulun opettajat eivät olleet päteviä ja erityisesti ohjelmointi-osuus tuotti monille opettajille vaikeuksia opettaa. Ohjelmointi ja järjestelmän aikainen kehittyminen koitui ongelmaksi opettajien pystyvyydessä. Nykypäivänä tietojenkäsittelytiede on muotoutunut oppiaineeksi keskittyen lähinnä digilukutaitoon peruskouluissa ja toisella asteella sisältäen sovellukset ja työkalut (kuten Office suite) ongelmanratkaisuun muissa aineissa. Kuitenkaan, ei ole olemassa tietojenkäsittelytiedettä oppiaineena pakollisissa kouluissa. Sen sijaan sisältö tarjotaan oppiaineissa kuten teknologia ja matematiikka. Toisen asteen koulutuksessa tietojenkäsittelytieteen koulutus vaihtelee riippuen koulutusohjelmasta johon opiskelija osallistuu, keskittyen luonnontieteisiin, teknologiaan, estetiikkaan tai sähköoppiin. Opettajien koulutus Ruotsissa ei tarjoa tietojenkäsittelytieteen kursseja, joten on epätavallista löytää opettajia, jotka opettaisivat näitä sisältöjä koulussa. Toisen asteen koulutuksessa on tarjolla kursseja ohjelmoinnista, multimediasta, verkkosivujen suunnittelua jne., mitä tarjotaan kelle tahansa opiskelijalle, joka haluaa opiskella näitä aiheita. Ohjelmointikursseille osallistuu lähinnä opiskelijoita yhdestä kolmesta koulutusohjelmasta teknologiassa, luonnontieteissä ja elektroniikassa. Tästä syystä opiskelijoista vain harva ottaa ohjelmointikursseja. Tieto- ja viestintäteknologian (ICT) sisällöt ruotsalaisessa koulutusjärjestelmässä ovat seuraavanlaisia:

- Esikoulun lukujärjestyksessä ainoastaan yksi tieto- ja viestintäteknologiaa koskeva tavoite liittyen haluttuun digitaalikompetenssiin kuuluu seuraavasti: "Multimediaa ja informaatioteknologiaa voidaan käyttää luovissa prosesseissa aivan kuten soveluksissa."
- Nykyisen pakollisen koulun (luokat 1 - 9) tieto- ja viestintäteknologian harjoittelu keskittyy suuntaamaan opiskelijoita "informaation flow'hun", kuinka käyttää teknologiaa, ja kehittämään kriittistä ajattelua internetistä löytyvää informaatiota kohtaan. Hiljattain Ruotsin hallitus on ehdottanut muutosta opetussuunnitelmaan, joka keskittyy enemmän digitaalisiin kompetensseihin.
- Koko Ruotsin nuoriso joka on suorittanut pakollisen oppimäärän on oikeutettu kolmivuotiseen toisen asteen koulutukseen (luokat 10 - 12 ja 13). Nykyään toisen asteen koulutus tarjoaa 18 opinto-ohjelmaa ja 61 suuntautumismahdollisuutta; 12 ammatilliseen työhön valmistavaa opinto-ohjelmaa ja kuusi opinto-ohjelmaa korkeampiin opintoihin, humanistisissa tieteissä, sosiaalitieteissä, luonnontieteissä, ja teknologian aloilla. Kolmastoista opiskeluvuosi on tarjolla vain teknologian koulutusohjelmassa erikoistumista varten yhdessä neljästä vaihtoehdosta. Toisen asteen koulutuksessa tietojenkäsittelytieteen kurssit ovat valinnaisia kaikille opiskelijoille. Informaatioteknologiaan liittyvät kurssit ovat pakollisia muutamissa kahdeksastoista koulutusohjelmasta, kuten teknologian koulutusohjelma sekä sähkö ja energia-koulutusohjelma. Valinnaisia ohjelmointikursseja on tarjolla kaikissa koulutusohjelmissa, mutta vain rajallinen määrä opiskelijoita aloittavat ohjelmoinnin koulussa.

On olemassa vapaaehtoisaloitteita ja epävirallisia ryhmiä (kuten esimerkiksi TeacherHack, Framtidens Språk), jotka toimivat parantaakseen nykyistä opetussuunnitelmaa, kehittämällä uusia ohjelmia ja innoittaen opettajia sisällyttämään enemmän CT:tä luokkahuoneessa. Opettaja-yhteisöt käyttävät sosiaalista mediaa (Twitteriä ja Facebookia) jakaakseen materiaaleja, ideoita ja levittämään sanaa siitä, kuinka kansallista opetussuunnitelmaa voitaisiin laajentaa esimerkiksi ohjelmoinnin avustuksella. Lisäksi ideat tekijä -liikkeestä innostavat opettajia tehdäkseen oppilaista ennemminkin luovia tekijöitä pelkkien tieto- ja viestintäteknologian kuluttajien sijaan. Ohjelmointia käytetään CT:n integrointiin oppiaineiden, kuten puusepäntöiden, ruotsinkielen ja teknologian kanssa.

Vuonna 2012 Ruotsin hallitus valitsi digitalisaation komitean (ruotsiksi: Digitaliseringskommisionen) antaakseen joitakin suosituksia digitalisaation työstämisen edistämiseksi ruotsalaisessa yhteisössä. Komitea julkaisi raportin, jossa korostetaan koulun tarvetta omaksua digitaalisen kompetenssin käsite, mikä on linjassa EU:n suositusten kanssa. Raportti tähdentää tarvetta ohjelmoinnin sisällyttämiseen opetussuunnitelmaan osana jo olemassa olevia oppiaineita.

3.1.5 Suomi

Lähteessä [Mannila, L. ym. (2014)] on esitelty, miten ohjelmallisen ajattelun (Computational Thinking, CT) opetusta on toteutettu Suomessa. Suomessa toisen asteen koulutuksen opetusussuunnitelmassa oli tietojenkäsittelytiede (Computer Science, CS) aina 2000-luvun alkuun asti [Mannila, L. ym. (2014)], mutta sittemmin siitä luovuttiin kokonaan. Sen sijaan tieto - ja viestintätekniikka (ICT) tuli osaksi kaikkia oppiaineita. Tämä johti tietokoneiden käyttämiseen lähinnä työkaluina, jolloin tietojenkäsittelytieteen näkökulma jäi varjoon, sillä opettajilla ei ollut taustaa saati kokemusta jälkimmäisen opettamisesta.

Nykyisin käytössä on uusi opetussuunnitelma, jossa äsken mainittuja osia on pyritty otamaan huomioon paremmin. Nykyisessä opetussuunnitelmassa korostetaan pääkäsitteiden ja periaatteiden ymmärtämistä, tieto - ja viestintätekniikan käyttämistä vastuullisesti ja turvallisesti, tieto - ja viestintätekniikan käyttämistä tiedon etsimiseen ja tieto - ja viestintätekniikan käyttämistä kommunikointiin ja yhteistyöhön. Näiden lisäksi Suomessa on ollut useita ohjelmointiin liittyviä aloitteita (initiatives) helpottamaan tätä reformia. Äsken mainittuihin aloitteisiin kuuluu Koodi2016 (opettajan opas); LUMA SUOMI -ohjelma, jossa opetus - ja kulttuuriministeriö tukee STEM aineita, joissa ohjelmointi on tärkeässä roolissa; opetushallituksen, joka on ollut päävastuussa tämän reformin aikaan saamiseksi, tuki opettajien ammatilliseen koulutukseen ja projekteihin, joissa osana on ohjelmointi.

3.1.6 Yhdysvallat

Yhdysvalloissa koulutuksen hallinnointi ja vastuu on ensisijaisesti valtion paikallisen hallituksen vastuulla, mikä tarkoittaa sitä, että opetussuunnitelmat ovat merkittävästi erilaisia kautta maan. Viimeisin askel kansallisen standardisoinnin suuntaan oli Yleiset ydinstandardit (Common Core Standards), jotka kehitettiin valtiijohtoisesti ja käynnistettiin vuonna 2009. Standardit on otettu käyttöön 44:ssä osavaltiossa, Kolumbian piirikunnassa, neljässä territoriossa, ja puolustuksen opetuksen toimen laitoksessa (Department of Defense Education Activity). Yleisissä ydinstandardeissa on kaksi joukkoa standardeja, jotka ulottuvat koko K - 12 opetussuunnitelmaan, yksi englannin kielen taiteelle tai kirjallisuudelle (literacy) ja yksi matematiikalle. On myös olemassa erillinen joukko standardeja luokille 6 - 12, jotka kattavat kirjallisuuden historiassa ja sosiaalisissa tieteissä, tieteessä ja teknisissä oppiaineissa.

Yleiset ydinstandardit sisältävät monia standardeja tietokoneelukutaidolle, enimmäkseen englantilaisen taiteen alueella. Esimerkiksi seuraavat ovat standardista löytyviä lukutaitoon liittyviä aiheita:

- Lukemisen perusstandardit: Yhdistää ja arvioida sisältöä, joka on esitetty monimuo-

toisissa medioissa ja formaateissa, mukaanlukien visuaalisesti ja kvantitatiivisesti.

- Kirjoittamisen perustandardit: Teknologian käyttäminen, mukaan lukien internet, kirjoitusten tuottamiseen ja julkaisuun ja tehdä yhteistyötä ja olla vuorovaikutuksessa muiden kanssa.
- Kirjoittamisen perustandardit: Kerätä olennaista tietoa useista painetuista ja digitaalisista lähteistä, arvioida jokaisen lähteen uskottavuutta ja tarkkuutta, ja yhteinäistää tämä tieto välttämällä plagiointia.

On myös olemassa joitakin aiheita, joita voidaan opettaa pitäen CT keskiössä, vaikkakin CT ei olekaan standardin päätavoite. Esimerkiksi seuraavat ovat pystyvyyyksiä, joita voitaisiin lähestyä CT:n kautta:

- Mittaus ja data: Esittää ja tulkita dataa (2. luokka): Piirtää kuvadiagrammi (picture graph) ja pylväsdiagrammi (käyttäen yhtä yksikköasteikkoa) esittääkseen datajoukko aina neljään kategoriaan asti. Ratkaista yksinkertaisia yhdistely-, erittely-, ja vertailu- ongelma tai pulmatehtäviä käyttäen tietoa pylväsdiagrammista.
- Ilmaisut ja yhtälöt: Käyttää operaatioiden ominaisuuksia tuottaakseen vastaavia ilmaisuja (7. luokka): Ymmärtää, että ilmaisun uudelleen kirjoittaminen eri muodoissa ongelmanratkaisun yhteydessä voi valaista ongelman ja sen suureiden välisten yhteyksien ymmärtämistä.

Yhdysvalloissa muutamassa paikassa on vaatimuksena tietojenkäsittelytiede opiskelijoille. Parhaimmillaan lukiossa (luokat 9 - 12) opiskelijoilla on vaihtoehto ottaa valinnaisena teknologiaa tai joissakin paikoissa edistynyt tietojenkäsittelytieteen (Advanced Placement Computer Science) kurssin, joka on tällä hetkellä ohjelmointikurssi Javalla. Kyseisellä kurssilla opiskelijat pääset osallistumaan standardoituun testiin, jonka läpipäästessään he ansaitsevat korkeakoulun opintopisteitä. Tyypillisesti opiskelijat ottavat vain tietojenkäsittelytiedettä, mikäli siitä on hyötyä valmistumisen kannalta, ja (joulukuuhun 2014 mennessä) vain 25 osavaltiota, joissa tietojenkäsittelytiede voidaan hyväksilukea tutkintoon. Riippuu osavaltiosta, mihin vaatimuksiin tietojenkäsittelytiede voidaan hyväksyä, mutta nykyiset oppiainekokonaisuudet ovat matematiikka, luonnontieteet, matematiikka ja luonnontieteet, matematiikka tai luonnontieteet sekä matematiikka ja muu. Yhdysvalloissa on käytössä useita eri yhdistelmiä oppiainekokonaisuuksista. Suuri osa aloitteesta saada tietojenkäsittelytiede myötävaikuttamaan valmistumiseen on tietojenkäsittelytieteen opettajien yhdistyksen (Computer Science Teachers Association, CSTA) ja hiljattain myös Code.org:in ansiota.

Ennen aiemmin mainittua aloitetta saada tietojenkäsittelytiede edistämään lukios-
ta valmistumista, oli tehty merkittäviä yrityksiä sekä monien järjestöjen että yksilöiden
osalta saada tarjottua vaihtoehtoja tietojenkäsittelytieteestä lukio-opiskelijoille, jotka va-
litsevat laajemman lähestymistavan tietojenkäsittelytieteeseen. Vaihtoehtoinen AP kurs-
si nimeltä *Tietojenkäsittelytieteen periaatteet* on kehitetty. Se on suunniteltu vetoamaan
opiskelijoihin, jotka eivät välttämättä ole kiinnostuneita ohjelmoinnista. Kurssi sisältää
muun muassa aiheita kuten abstraktio, big data ja informaatio, algoritmit, ja internet.
Tietojenkäsittelytieteen periaatteet AP koe on alustavasti suunniteltu tarjottavaksi ensi
kertaa toukokuussa 2016.

Ei ole olemassa näyttöä tai todisteita, että mikään perus- tai yläkoulu vaatisi tietojen-
käsittelytieteen kursseja valmistumiseen. Code.org, CSTA ja tietojenkäsittelytieteen
opetuksen tutkijat ovat työstäneet muuttaakseen tilannetta, sekä kehittämällä opetusmate-
riaaleja että kouluttamalla opettajia.

Yhdysvalloissa opettajalla täytyy olla kandidaatintutkinto (neljän vuoden yliopisto-
tutkinto) ja täytyy olla ansaittua osavaltion myöntämä sertifikaatti tai lisenssi. Jotkut
osavaltiot vaativat lisäksi alakoulun opettajilta syventäviä opintoja jostain sisältöalueesta
kuten matematiikka tai luonnontieteet. He ilmoittautuvat yliopiston opettajille tarkoitettuun
valmistavaan ohjelmaan ja valitsevat kursseja opetuksesta ja lasten psykologiasta,
niiden kurssien lisäksi, jotka kuuluvat pääaineeseen. Huomautettakoon, että yksityiskou-
lujen opettajat eivät tarvitse aiemmin mainittuja osavaltion vaatimuksia, kuten sertifikaatteja
ja lisenssejä.

Vuonna 2007 CSTA keräsi tietoa sertifikaateista ja hyväksymismerkinnöistä tietojen-
käsittelytieteen opettajille 50:stä osavaltiosta. Neljäkymmentäseitsemän osavaltiota pa-
lautti vastauksensa, joiden vastausprosentti oli 88%. Kysyttäessä myöntävätkö he tietojen-
käsittelytieteestä hyväksymismerkintöjä, vain 53 % osavaltioista ilmoitti tekevänsä
näin. Kuitenkin vastaukset osoittivat, että 29 % vaativat tietojenkäsittelytieteestä hyväk-
symismerkintää toisella asteella, 27 % vaatii sitä yläkoulussa, 13 % vaatii sitä alakoulun
tasolla, ja 13 % vaatii sitä koko K - 12 asteella. Joissakin osavaltioissa opettajien sallittai-
siin opettaa tietojenkäsittelytieteen kursseja käyttäen muita sertifikaatteja, mukaan lukien
matematiikka, opetuksellinen teknologia, teknologia ja muotoilu (design), ja liiketoiminta
(business) sekä tieto- ja kirjastopalvelut (library media).

Kuulu opettajien sertifikaattien tai hyväksymismerkintöjen saatavuuden ja vaatimus-
ten, että opettajilla on hallussaan sertifikaatteja tai hyväksymismerkintöjä opettaa tietojen-
käsittelytieteen aiheita, välillä, on suuri huoli Yhdysvalloissa. Kansallinen tietesäätiö
(The National Science Foundation) on usean vuoden ajan rahoittanut projekteja, jotka
keskittyvät opettajien kehittämiseen tietojenkäsittelytieteessä CS 10K ohjelmassa, joka
tällä hetkellä on osa STEM-C (partnerships program) kumppanuusohjelmaa. Ohjelman
tavoite on kouluttaa 10 000 tietojenkäsittelytieteen opettajaa Amerikkalaisessa koulujär-
jestelmässä.

3.2 Ohjelmoinnin opettamisesta muualla

Käsitellään seuraavaksi kuinka ohjelmointia on opetettu vuonna 2015 muualla Euroopassa [European Schoolnet (2015)]. Tämä luku on tutkielman kannalta hyvin mielenkiintoinen, sillä tämän tutkielman mielenkiinnon kohteena oli erityisesti ohjelmoinnin opetuksen sijoittaminen opteussuunnitelmassa. Siinä missä muissa maissa ohjelmoinnin opetus voi riippua hyvinkin suuresti jopa maan sisällä erinäisistä alueiden tai koulujen omien omista opetussuunnitelmista, on Suomessa päätetty painottaa ohjelmoinnin opetus matematiikan opetukseen. On myös kiinnostavaa nähdä muita vaihtoehtoisia tapoja maailmalta ohjelmoinnin opetukseen.

3.2.1 Bulgaria

Bulgariassa "Informatiikka" on pakollinen oppiaine 9. luokalla. Oppilailla on kaksi tuntia opetusta per viikko. Kurssilla opetetaan perustiedot (basic knowledge) tietojenkäsittelytieteen käsitteistä ja matemaattisia periaatteita. Tämän lisäksi kurssilla opetetaan ohjelmointia, ongelmanratkontaa algoritmeilla ja tiedon esittämistä abstraktion (kuten mallien ja simulaatioiden) avulla, ovat osa molempia oppiaineita: informatiikka ja ICT.

3.2.2 Espanja

Kansallisella tasolla koodausta tarjotaan valinnaisena aineena toisen asteen koulutuksessa oppiaineessa nimeltä "Tecnologías de la Información y la Comunicación", joka kääntyy suomeksi "Informaatio- ja kommunikointiteknologiat". Kuitenkin paikallisia toteutuksia on myös käytössä ohjelmoinnin opetuksessa.

- Autonomisessa yhteisössä Madridissa, robotiikkaan ja koodaukseen liittyvät sisällöt on sisällytetty yleiseen oppiaineeseen "Tecnología, Programación y Robótica", joka on yläkoulun kolmannen vuoden yleinen teknonologian oppaine.

3.2.3 Irlanti

Toisen asteen koulutuksessa valinnainen lyhyt kurssi koodauksesta on otettu käyttöön 13 - 15 -vuotiaille suunnatussa *junior cycle programme* -ohjelmassa. Ongelmanratkonta (algoritmisen ajattelun) CT:n taitoja kehitetään kurssilla kun oppilaat rakentavat ja luovat ohjelmistoprojekteja käyttäen omia ideoitaan ja mielikuvitustaan. Kurssi pyrkii tavoittamaan opiskelijat, jotka saattavat olla kiinnostuneita jatkokoulutuksesta tietojenkäsittelytieteissä tai ohjelmiston tuottamisesta (software engineering).

3.2.4 Israel

Israelin osalta ei ole mainittu tarkempia tietoja, ainoastaan tiedot, mitä tietojenkäsittelytiede oppiaineena sisältää. Myöskään ei ole mainintaa mitä luokka-asteita tarkoitetaan. Seuraavat sisällöt voivat koskea esimerkiksi yläkoulua tai lukiota, tai jopa vieläkin korkeampaa astetta:

- algoritmien ajattelu,
- luovuus,
- ongelmanratkenta,
- automatiikka, perustiedot käyttöjärjestelmistä,
- C# -ohjelmointikieli, Java & assemblit.

3.2.5 Itävalta

Itävallassa koodausta opetetaan osana kokeiluluontoisia ohjelmia (school trials) yläkouluissa fokuksena informatiikka (Informatics), web-pohjaisella koodauksella ja toisella ohjelmointikielellä. Sekä yläkouluissa että lukioissa, joissa informatiikka on keskiössä, koodausta opetetaan kahdella ohjelmointikielellä. Näissä kouluissa se, onko koodaus integroitu yleiseen ICT/teknologia kursseihin, riippuu alueellisesta tai koulun omasta opetussuunnitelmasta.

3.2.6 Malta

Maltalla ohjelmoinnin opetus on koki muutoksia, kun uusi oppimistulosten kehys (Learning Outcomes Framework) määritteli kaikkien oppiaineiden fokuksen, mukaan lukien tietojenkäsittelyn. Perustelu muutoksille aiempaan nähden on "tietojenkäsittely oppiaineen" laajuus verrattuna aiempaan toteutukseen opetussuunnitelmassa. Tietojenkäsittely sisällytettiin pakolliseen toisen asteen ICT oppaineeseen.

3.2.7 Puola

Vuonna 2016 Puolassa otettiin käyttöön uusi tietojenkäsittelytieteen opetussuunnitelma, joka korvasi jotkin aktiviteetit informaatioteknologian sisällä opettamalla informaatioteknologian sijaan täsmällistä tietojenkäsittelytiedettä, mukaan lukien ohjelmointi. Uusi opetussuunnitelma yhdistää kaikki itsenäiset oppiainetoteutukset informatiikassa kulkemaan

jatkossa nimellä Informatiikka. Tällöin myös uuden opetussuunnitelman myötä informatiikasta tuli pakollinen oppiaine alakoulussa (1. - 6. luokat), yläkoulussa (luokat 7. - 9. luokat). Alakoulussa oppiainetta on yksi vuosiviikkotunti kuuden vuoden ajan. Yläkoulussa oppiainetta on yksi vuosiviikkotunti kahden vuoden ajan. Lukiossa on myös mahdollista opiskella informatiikkaa pakollisen yhden vuosiviikkotunnin lisäksi enemmän.

Tiivistetysti uuden opetussuunnitelman tuomat yhtenäistetyt tavoitteet ovat:

1. Pulmien ja ongelmien ymmärtäminen ja analysointi.
2. Ohjelmointi ja ongelmanratkenta käyttäen tietokoneita ja muita digitaalisia laitteita.
3. Tietokoneiden, digitaalisten laitteiden ja tietoverkkojen käyttö.
4. Sosiaalisten kompetenssien kehittäminen.
5. Laki- ja turvallisuusperiaatteiden ja -säästösten havainnointi.

3.2.8 Slovakia

Algoritmista ajattelua tai tietokoneohjelmointia kehitetään oppiaineessa nimeltä "Informatiikka" tai muiden oppituntien aikana. Yläkouluissa opettajat käyttävät opetuksellisia alkeellisia ja graafisesti suuntautuneita visuaalisia ohjelmointityökaluja lapsille, kuten "Karel", "Baltie" tai "Imagine". Opettajat käyttävät myös ohjelmoitavissa olevien robottien rakennussarjoja.

3.2.9 Tanska

Tanskassa 7. - 10. luokilla koodaus on sisällytetty sitovaan ja velvoittavaan kansallisiin yhteisiin tavoitteisiin (Common Objectives) fysiikalle ja kemialle. Tavoitteita ovat:

- tietämys (knowledge) yksinkertaisesta ohjelmoinnista ja datan siirrosta,
- ohjelmointikielet ja taito ohjelmoida yksinkertaisia digitaalisia ratkaisuja (fysiikka ja kemia),
- matematiikassa systemaattisen ja abstraktin ajattelun vahvistaminen erityisellä ohjauksella.

3.2.10 Tsekin tasavalta

Toisen asteen koulutuksen (yläkoulu ja lukio) koulutuksellisten opinto-ohjelmien kehys sisältää (FEP = Framework Educational Programmes) (Suomessa vastaava maan opetussuunnitelma) perusteet tietojenkäsittelystä tieteenalana ja koulutuksellisesta alaa kutsutaan tietojenkäsittely ja informaatio ja kommunikointiteknologioiksi (Computing and Information and Communication Technologies). Tämä tarkoittaa sitä, että kaikkien toisen asteen koulujen täytyy sisällyttää ohjelmointia koulun omaan opetussuunnitelmaan (SEP = School Education Programme) - mutta aika, joka omistetaan ohjelmoinnin opettamiselle, riippuu täysin koulusta.

Koodauksen (ohjelmoinnin) perusteet, tai yleisemmin algoritmien ajattelu ja CT saattavat muodostaa jonkin laajemman osan ICT:n oppiaineesta. Jotkin koulut tarjoavat ohjelmointia valinnaisena oppiaineena. Opetettava aines ja valinnaisten oppiaineiden nimet ovat koulujen päätettävissä ja vaihtelevat suuresti.

3.2.11 Unkari

Unkarissa vuodesta 2012 alkaen kehysopetussuunnitelma (the Frame Curricula) odottaa käytettävän ICT:tä eri oppiaineissa pois lukien koodaus. Vaikkakin informatiikka on omana oppiaineenaan, koulut voivat valita sen sisällyttämisen oppitunnin tarkoituksen mukaan.

- 13 - 14 -vuotiaat: 1) Algoritmit 2) Logo tai toinen yksinkertainen ohjelmointikieli, 3) Perus ohjelmointikäsky,
- 15 - 16 -vuotiaat: 1) Algoritmin suunnittelu ja analyysi, 2) Ongelmanratkenta.

3.2.12 Yhdistynyt kuningaskunta (Englanti)

Englannissa tietojenkäsittely on omana oppiaineenaan koulun opetussuunnitelmassa mutta koulut ovat vapaita opettamaan sitä toiseen oppiaineeseen integroituna oppiaineena tai erikseen. Alakouluissa on yleisempää opettaa tietojenkäsittelyä integroituna muihin oppiaineisiin verrattuna toisen asteen koulutukseen, jossa oppiainerajat ylittävä työskentely on harvinaisempaa. Englannin kansallisessa opetussuunnitelmassa tietojenkäsittelyn tavoitteena on, että oppilaat:

- ymmärtävät ja soveltavat perustavanlaatuisia periaatteita ja käsitteitä tietojenkäsittelytieteestä, mukaan lukien abstraktio, logiikka, algoritmit ja datan esittäminen,
- osaavat analysoida ongelmia tietokoneiden avulla ja omaavat käytännön kokemusta tietokoneohjelmien kirjoittamisesta ratkaistakseen kyseisiä ongelmia,

- osaavat arvioida ja soveltaa informaatioteknologiaa, mukaan lukien oppilaalle uudet ja ennestään tuntemattomat teknologiat, analyyttisesti ratkaistakseen ongelmia,
- ovat vastuullisia, kykeneviä, varmoja ja luovia informaatio- ja kommunikointiteknologian käyttäjiä.

3.3 Tutkimusmenetelmänä kyselytutkimus

Tässä tutkielmassa käytettiin tutkimusmenetelmänä kyselytutkimusta, joka koostui yhdeksästätoista kysymyksestä ja viimeisenä oli "vapaa sana" kenttä, johon sai tavan mukaan kertoa ajatuksiaan tutkimukseen liittyvistä aiheista. Kysymykset oli jaoteltu kolmeen osioon. Ensimmäisessä osassa pyrittiin kartoittamaan opettajien taustoja, kuten työkokemusta. Ensimmäisessä osassa kartoitettiin myös opettajien asenteita ohjelmointia ja sen opetusta kohtaan. Kyselytutkimuksen toisessa osassa kysyttiin opettajien tavoista toteuttaa ohjelmointia matematiikan opetuksessa. Lisäksi toisessa osassa kartoitettiin opettajien käytössä olleita resursseja sekä mm. opettajan käyttämiä oppimateriaaleja. Kolmannessa osassa kartoitettiin opettajan ohjelmointitaitoa kysymyksillä, jotka koskivat ohjelmoinnin pedagogiikkaa ja opettajan omaa ohjelmointikokemusta sekä koulutukseen osallistumisesta. Kyselytutkimuksen kohderyhmänä olivat aineenopettajat, jotka opettavat matematiikkaa yläkoulussa.

Kyselytutkimuksen tuloksia analysoidaan tilastollisten menetelmien avulla. Tutkimusmenetelmä on luonteeltaan empiirinen, eli saadaan jonkinlainen käsitys todellisuuden nykytilasta tutkittavan aiheen näkökulmasta. Kyselytutkimus koettiin hyväksi tavaksi tutkia tutkittavaa ilmiötä, sillä kyselytutkimuksessa saadaan selvitettyä mielekkäästi suuren joukon mielipiteitä. Tässä tutkielmassa mielenkiinnon kohteena olivat mielipiteet ohjelmoinnin opetuksesta matematiikan opetuksessa. Kyseessä on otantatutkimus, sillä tämän tutkielman tekemiseen varatussa ajassa olisi ollut äärimmäisen vaikeaa saada kaikki yläkoulun matematiikkaa opettavat vastaamaan kyselytutkimukseen. Lisäksi sopivan suurella otannalla saadaan tarpeeksi kattava otos koko populaatiosta.

3.4 Kyselytutkimus

Kuten aiemmin on jo mainittu, oli kyselytutkimuksessa yhdeksäntoista kysymystä. Nämä kysymykset käydään läpi seuraavaksi läpi, ja annetaan samalla niille perustelu, miksi juuri ne on valittu tähän kyselytutkimukseen. Annettavat perustelut valottavat tutkielman näkökulmaa, ja sitä, mitä tutkielmalla on haluttu tutkia. Tätä tutkielmaa varten tehty kyselytutkimus toteutettiin vuoden 2018 marras - joulukuussa. Kyselyyn vastaajat

täyttivät kyselylomakkeen sähköisesti Helsingin yliopiston E-lomake -palvelussa. Kaikki kyselylomakkeiden vastaukset säilytettiin sähköisesti kyseisessä palvelussa.

3.4.1 Ensimmäinen osio: opettajien omat taustat ja asenteet

Ensimmäisessä kysymyksessä kysyttiin vastaajan sukupuolta, johon vastaaja sai vastata "Mies", "Nainen" tai "Muu/En halua vastata". Kysymyksen ensisijainen tarkoitus on toimia taustamuuttujana, jotta voidaan verrata, miten muiden kysymysten vastaukset ovat sijoittuneet miesten, naisten ja muiden kesken. Samalla havaitaan, onko tällä tavoin lajiteltujen ryhmien vastauksien välillä huomattavia eroja.

Toisen kysymyksen avulla taustoitettiin vastaajan työkokemusta vuosissa matematiikan opettajana. Vastaajan ikää merkittävämpänä tekijänä opettamisessa koettiin olevan vastaajan työkokemus matematiikan opettajana. Ensinnäkään vastaajan ikä ei korreloi suoraan vastaajan taitoja saatika kykyjä matematiikan opettajana. Tästä syystä vastaajan työkokemus koettiin merkittävämmäksi tekijäksi, kun haluttiin selvittää vaikuttaako työkokemus joko positiivisesti tai negatiivisesti ohjelmoinnin opettamisen toteutukseen matematiikan opetuksessa. Ensimmäisenä vastausvaihtoehtona oli 0 - 5 vuotta, ja tästä kokemusmäärästä käytetään ilmaisua *vähäiseksi kokemukseksi* tässä tutkielmassa. Tähän kategoriaan kuuluvat ne opettajat, jotka ovat vasta hiljattain valmistuneet yliopistosta sekä ne opettajat, joilla on verrattain melko vähän opetuskokemusta. Lisäksi tähän kategoriaan kuuluvat ne opettajat, jotka ovat mahdollisesti saaneet tuoreinta tietoa ohjelmoinnin opettamisesta. Toisena vastausvaihtoehtona oli 6 - 15 vuotta, ja tästä kokemusmäärästä käytetään ilmaisua *kohtalaiseksi kokemukseksi*. Tässä kategoriassa ovat ne opettajat, jotka ovat olleet opettajana jo sen verran, että heille on kerääntynyt kokemusta ainakin kahdesta opetussuunnitelmasta. Opettajilla alkaa olla jo vankka näkemys siitä, kuinka oppilaat oppivat parhaiten. Kolmantena vastausvaihtoehtona oli 16 - 29 vuotta, ja tästä kokemusmäärästä käytetään termiä *vahvaksi kokemukseksi*. Tähän kategoriaan kuuluvat ne opettajat, jotka ovat kokeneet useat opetussuunnitelmat ja miten opetuksessa käytetyt eri ilmiöt ovat näkyneet vuosien varrella. Neljäs ja viimeinen vastausvaihtoehto oli Yli 30 vuotta, ja tästä kokemusmäärästä käytetään ilmaisua *erittäin vahvaksi kokemukseksi*. Tässä kategoriassa ovat ne opettajat, jotka ovat nähneet matematiikan opetuksen useat eri vaiheet kuten esimerkiksi tasoryhmät.

Kolmannen kysymyksen avulla kartoitettiin vastaajan asennetta ohjelmointia kohtaan. Vastausvaihtoehtoja oli viisi, vaihdellen vaihtoehdosta "En ole lainkaan kiinnostunut" aina "Olen erittäin kiinnostunut" vaihtoehtoon. Tutkimuksessa selvitetään matematiikan opettajan omaa asennetta ja samalla mahdollista motivaatiota ohjelmointia kohtaan, sillä opettajan oma asenne vaikuttanee merkittävästi oppilaiden mielenkiintoon, mikäli he havaitsevat, että opettaja ei ole itse kovin kiinnostunut aiheesta.

Neljännessä kysymyksessä selvitettiin opettajan asennetta ohjelmoinnin opetusta koh-

taan. Vastausvaihtoehtoja oli tässäkin kysymyksessä viisi kappaletta, ja samoin kuin edellisessä kysymyksessä, vaihtoehdot olivat muotoa "En ole lainkaan kiinnostunut" ollen lievin kiinnostuksen muoto ja vastauksen "Olen erittäin kiinnostunut" olevan vahvin kiinnostuksen muoto. Kysymys on merkittävä, sillä mikäli opettajalla ei ole intoa ohjelmoinnin opetusta kohtaan, on mahdollista, että opettaja toteuttaa ohjelmointia matematiikan opetuksessa vain ja ainoastaan esimerkiksi paperilla ja kynällä varsinaisten ohjelmointiympäristöjen sijaan. Mainittakoon lisäksi se, että vielä edellisessä, vuoden 2004 Perusopetuksen opetussuunnitelmassa ei ollut yhtään mainintaa ohjelmoinnista [OPS 2004]. Näin ollen vielä 15 vuotta sitten opiskelleet matematiikan opettajat eivät voineet tietää, että heidän tulee opettaa matematiikan lisäksi ohjelmointia matematiikan opetuksessaan. Tämä tulee mahdollisesti näkymään kokeneempien vastaajien vastauksissa.

Viidennessä kysymyksessä haluttiin kuulla vastaajan näkemys siitä, miten ohjelmoinnin opetusta matematiikassa pitäisi toteuttaa. Vaihtoehtoina oli "Microsoft Excelin / LibreOffice Calcin opettaminen riittää", "Oppilaiden pitäisi oppia algoritmista / ohjelmallista ajattelua", "Oppilaiden pitäisi oppia kirjoittamaan yksinkertaista koodia" ja "Muu (Miten?)". Näistä vaihtoehdoista viimeiseen yhteyteen oli varattu tekstikenttä vastaajan vapaamuotoiselle vastaukselle. Ensimmäinen vaihtoehto pyrki kuvaamaan hyvin yksinkertaista funktionaalista ohjelmointia, esimerkiksi Excelissä on valmiina funktio nimeltä Summa(), jonka argumentteina on solujen arvot, jotka halutaan summata yhteen. Näin opettamalla oppilaiden yksinkertainen koodinlukutaidot saattavat kehittyä, mutta varsinaiset ohjelmointitaidot jäänevät hyvin vähäisiksi. Toisen vaihtoehdon avulla pyrittiin tavoitella tilannetta, jossa oppilas oppisi ymmärtämään ajatuksen tasolla mistä vaiheista kokonainen ohjelma koostuu. Tässä tilanteessa ei tarvitse osata välttämättä ohjelmoida, vaan tärkeämpää olisi itse ymmärrys siitä, miten algoritmit (analogiana resepti) muodostuvat. Kolmas vaihtoehto pyrki kuvaamaan tilannetta, jossa oppilas oppisi jo kirjoittamaan yksinkertaista koodia. Yksinkertaisena koodina tässä pidetään ohjelmaa, jossa voitaisiin laskea esimerkiksi ohjelmalle annettavan kahden luvun summa, tai vaikkapa ohjelmalle annettavan kokonaisluvusta koostuvasta listasta laskettavan lukujen keskiarvon. Viimeisenä vaihtoehtona oleva "Muu (Miten?)" pyrki muotoilunsa mukaan juurikin selvittämään, miten muuten ohjelmointia voisi opettaa matematiikan opetuksessa.

Kuudennen ja samalla ensimmäisen osion viimeisellä kysymyksellä selvitettiin vastaajan mielipidettä siitä, tarvitsevatko kaikki oppilaat ohjelmoinnin opetusta. Kysymyksen tarkoituksena oli selvittää kokevatko opettajat ohjelmoinnin opetuksen kaikille turhana vaiko tarpeellisenä. Tutkielman laatijan näkemys asiasta on se, että ohjelmoinnin opetuksella on annettavaa kaikille. Mikäli oppilas opiskelee ohjelmointia, hänen ajattelutaitonsa kehittyvät, aivan kuten matematiikassakin. Ohjelmoinnin opettamisen tai opiskelun ei tulisi olla pelkkää koodin kirjoittamista, eikä matematiikan opettamisen tai opiskelun tulisi olla pelkkää luvuilla laskemista.

3.4.2 Toinen osio: opettajien resurssit, ohjelmoinnin opettamisen toteuttaminen ja mielipiteitä ohjelmoinnin opettamisesta

Toisen osion ensimmäinen kysymys ja samalla kyselylomakkeen seitsemäs kysymys koski sitä, miten vastaaja on toteuttanut ohjelmointia matematiikan opetuksessa. Tähän kysymykseen ei ollut esiannettuja vastausvaihtoehtoja. Vastaaajalta haluttiin kuulla sekä toimintatavat joilla hän on toteuttanut ohjelmointia että mitä hän itseasiassa pitää ohjelmoinnin toteuttamisena. Tähän kysymykseen voidaan odottaa hyvinkin erilaisia vastauksia, sillä vastaajien näkemykset ohjelmoinnin toteuttamisesta saattavat hyvinkin erota toisistaan. Joku voi pitää ohjelmointina vain ja ainoastaan varsinaisessa ohjelmointiympäristössä (kuten Netbeans [Netbeans]) tehtävää ohjelmointia, toisaalta joku muu voi hyväksyä ohjelmoinnin olevan aiemman lisäksi esimerkiksi algoritmillisen ajattelun harjoittelua tai vaikkapa pelaamalla ohjelmointiin liittyviä pelejä (kuten code.org [code.org]). Tarkkaa määritelmää opetussuunnitelman perusteissa ei anneta ohjelmoinnille, joten luonnollisesti opettajien omat mielikuvat ja määritelmät näyttelevät merkittävää roolia tähän kysymykseen vastatessa.

Kahdeksannessa kysymyksessä kysyttiin ohjelmoinnin opettamisen ajoittamisesta matematiikan opetuksessa. Kysymys laadittiin siitä syystä, että vastaajien keskuudessa lienee merkittäviä eroja valita ajankohta toteuttamiselle. Jotkut vastaajat voivat mahdollisesti ajoittaa ohjelmoinnin opetuksen joko yhden tai jokaisen lukuvuoden viimeisille viikoille. Tämä tuskin lienee ollut opetussuunnitelman perusteiden tavoite, vaan ennemminkin parempi vaihtoehto lienee toteuttaa esimerkiksi lyhyitä opetustuokioita pitkin lukuvuotta, jolloin oppilailla olisi pidempi aika omaksua uusi asia. Jotkut vastaajat mahdollisesti saattavatkin toteuttaa ohjelmointia matematiikan opetuksessaan juuri näin.

Yhdeksännessä kysymyksessä selvitettiin kuinka vastaaja on toteuttanut ohjelmointia matematiikan opetuksessa vastausvaihtoehtojen avulla. Vastaustavaihtoehdot olivat seuraavat: "Kynällä ja paperilla.", "Tietokoneella." ja "Muulla tavalla, miten?". Viimeisen vaihtoehdon jälkeen oli vastauskenttä, johon sai tarkentaa vastausta. Kysymyksen tavoitteena oli seitsemännen kysymyksen tapaan selvittää tapoja, joilla matematiikan opettajat toteuttavat ohjelmointia opetuksessaan. Yhdeksännessä kysymyksessä haluttiin kuitenkin ohjata vastaajaa pohtimaan, noudattaako hän vain jompaa kumpaa tapaa kahdesta ensimmäisestä yksinkertaisesta tavasta, vai ymmärtääkö hän ohjelmoinnin olevan laajempi käsite.

Kymmenennessä kysymyksessä selvitettiin vastaajan kokemusta siitä, kuinka paljon hänellä on ollut tiettyjä resursseja käytössään ohjelmoinnin opetusta toteuttaessa matematiikan opetuksessa. Kysymys jakautui alakohtiin resurssien mukaan, mutta kaikissa alakohdissa oli samat seuraavat vastausvaihtoehdot: "Ei ollenkaan.", "Liian vähän.", "Melkein riittävästi." ja "Riittävästi.". Aiemmin mainitut tietyt resurssit olivat "Tietokoneita", "Oppimateriaaleja" ja "Aikaa". Kysymyksen tarkoitus oli selvittää, onko opettajilla käy-

tössään tarpeeksi resursseja opettaa ohjelmointia, mikäli näin ei ole, olisi siihen vaikuttavien tahojen harkita jatkotoimenpiteitä. Lienee täysin ymmärrettävää, että mikäli opettajalta puuttuu jotain näistä mainituista resursseista, voi opetuksen laatu kärsiä huomattavasti. Suomessa yläkoulut saavat varansa pääosin kunnalta [Opetus- ja kulttuuriministeriö], täten lienee täysin perusteltua olettaa, että koulun maantieteellinen sijainti vaikuttaa merkittävästi resursseihin, joita koululla on käytössä. Tämä olisikin ollut mielenkiintoinen lisäkysymys kyselylomakkeeseen, mutta valitettavasti se ei selvinnyt kyselylomakkeeseen asti. Tällä tavoin kuitenkin saadaan suojeltua paremmin vastaajien anonymiteettiä, sillä mikäli tarpeeksi pienessä paikkakunnassa opettajana toimiva henkilö olisi vastannut kyselyyn, olisi täysin mahdollista selvittää opettajan henkilöllisyys. Tämä taas olisi voinut vaikuttaa opettajien mahdollisesti kritiikkiä sisältäviin vastauksiin, jolloin vastaaja ei olisi uskaltanut kertoa omaa todellista kantaansa. Nykypäivänä on melko yleistä (ainakin pääkaupunkiseudulla), että koulussa oppilaiden käytössä on joko kannettava tietokone, joka on mahdollista tuoda oppitunnille osana tavallista oppituntia, tai pöytäkone, joka sijaitsee tietokoneluokassa. Erityisen mielenkiintoista on tietää, ovatko opettajat löytäneet laadukkaita materiaaleja opetuksen tueksi. On myös mahdollista, että suomeksi ei ole saatavilla kyseisiä materiaaleja, vaan niitä toivottaisiin jopa tuotettavan lisää. Viimeinen alakohta koski aikaa resurssina. Mikäli opettaja ei löydä riittävästi aikaa ohjelmoinnin opettamisen toteuttamiseen, saattaa opettaja toteuttaa ohjelmoinnin opetuksen joko hätiköiden, tai pahimmassa tapauksessa jättää kokonaan toteuttamatta.

Yhdennentoista kysymyksen avulla kartoitettiin vastaajan tyytyväisyyttä siihen, miten ohjelmointia toteutetaan matematiikan opetuksessa tällä hetkellä vastaajan omassa koulussa. Vastausvaihtoehtoja oli neljä, joiden avulla kartoitettiin vastaajan tyytyväisyyttä nykyiseen toteutustapaan sekä mielipidettä siitä, onko tarvetta kehittää nykyistä toteutustapaa. Kysymyksen avulla saadaan tietoa siitä, ovatko vastaajat yleisesti ottaen tyytyväisiä koulunsa ohjelmoinnin opetukseen. Kysymystä tarkemmin analysoitaessa tulee ottaa huomioon mahdollisuus, että vastaaja on koulunsa ainoa matematiikan opettaja. Vastaukset tähän kysymykseen antavat vain suuntaa siitä, minkälaisena opettajat kokevat ohjelmoinnin toteutuksen tason olevan koulussaan.

Kahdestoista kysymys koski ohjelmoinnin opettamisen toteuttamista yhteistyönä muissa aineissa, kuten esimerkiksi fysiikassa. Kysymyksellä haluttiin selvittää ensiksikin miten yleistä yhteistyön toteuttaminen ohjelmointia opettaessa on. Toiseksi oltiin kiinnostuttu siitä, millä tavalla mahdollinen yhteistyön toteutus on tapahtunut ja minkälaisia yhteistyöprojekteja on toteutettu. Fysiikan roolia korostettiin kysymyksenasettelussa siitä syystä, että yksi luontevimmista ohjelmoinnin sovelluskohteista yläkoulussa voi olla juurikin fysiikka. Fysiikan lisäksi yhteistyötä voi tietysti toteuttaa muissakin oppiaineissa, mutta nämä lienevät hieman epätavallisempia. Tästä syystä onkin mielenkiintoista kuulla, miten yleistä on toteuttaa yhteistyötä muissa aineissa, sekä mitä erilaisia projekteja on näissä oppiaineissa toteutettu.

Kolmannessatoista kysymyksessä taustoitettiin sitä, onko oppilailla ollut keskimäärin riittävät esitiedot ohjelmoinnista. Vastausvaihtoehdot olivat vain "Kyllä." ja "Ei.", muita vaihtoehtoja ei katsottu tarpeellisiksi, sillä kyse oli koko oppilasjoukosta ja samalla taitojen riittävydestä. Kysymyksen tarkoituksena oli taustoittaa, ovatko opettajat kokeneet oppilaiden esitiedot olleet riittävän hyvät, jotta ohjelmointia olisi ollut mielekästä opettaa. Vaikkakin ohjelmointi on monelle uusi asia yläkoulussa, tarvitaan siinä tietynlaista ajattelumallia. Siitä huolimatta että eritasoisia tehtäviä ja materiaaleja löytyy ohjelmoinnin opetukseen ja opiskeluun, on mahdollista että jotkin oppilaat eivät saa selkeää käsitystä siitä, mitä ohjelmoinnissa oikeastaan tehdään. Tämä voi näkyä opettajalle myös oppilaan haasteissa matematiikan parissa, sillä samanlaista ajattelumallia tarvitaan molemmissa.

Neljännessätoista kysymyksessä kysyttiin, onko vastaajalla ollut käytössä hyviä oppimateriaaleja ohjelmoinnin opetuksessa. Vastata pystyi yhdellä kahdesta vastausvaihtoehdosta: "Ei." ja "Kyllä (Mikä/Mitkä?).", joista jälkimmäisen vastattuaan vastaaja saattoi tarkentaa täytettävään tekstikenttään. Ensimmäisessä kysymyksellä haluttiin kartoittaa ohjelmoinnin opetuksessa käytettäviä materiaaleja, joita opettajat ovat mahdollisesti käyttäneet opetuksessaan. Toissijaisesti kysymyksen tarkoitus oli vastata kysymykseen, ovatko opettajat löytäneet tarkoitukseen soveltuvia materiaaleja. Mikäli soveltuvia materiaaleja ei ole löytynyt, on opettaja mahdollisesti tehnyt itse omat materiaalisensa.

Viidennessätoista kysymyksessä haluttiin kuulla mitä ohjelmointikieliä vastaaja on käyttänyt toteuttaessaan ohjelmoinnin opetusta. Vastausvaihtoehtoina oli "Pythonilla.", "Javalla.", "JavaScriptillä.", "C++ :lla.", "Jollain muulla." ja "En millään". Vastaajan oli valittava näistä vastausvaihtoehdoista vähintään yksi. Kysymyksen avulla haluttiin selvittää, käyttävätkö opettajat ohjelmoinnin opetukseen jotain suosituimmista ohjelmointikielistä vai jotain muuta ohjelmointikieltä. Vastaajalle annettiin myös vaihtoehto, jossa hän ei ole toteuttanut ohjelmointia millään ohjelmointikiellä. Tämä ei kuitenkaan estä ohjelmoinnin opettamista, sillä esimerkiksi algoritmista ajattelua voi opettaa ilman minkään ohjelmointikielen käyttämistä.

Toisen osion päätti kysymys "Onko koulussanne toteutettu ohjelmoinnin opetus matematiikassa vai pidetäänkö kurssi erikseen?", joka oli järjestyksessään kuudestoista. Vastausvaihtoehdot olivat seuraavat: "Ohjelmoinnin opetus on toteutettu matematiikassa", "Kurssi pidetään erikseen.", "Muut (Miten?)." ja "Ei ole toteutettu.". Kysymyksellä haluttiin selvittää, missä määrin Suomen yläkouluissa ohjelmoinnin opetus toteutetaan matematiikan opetuksessa ja missä määrin muilla tavoin. Vastaajat, jotka valitsivat ensimmäisen vastausvaihtoehdon ovat toteuttaneet ohjelmoinnin opetusta juuri kuten opetussuunnitelman perusteissa on toivottu. Tällä tavalla saadaan yhteys matematiikkaan vahvistettua ja ohjelmoinnin opetus motivoitua paremmin. Toisen vastausvaihtoehdon valinneet vastaajat ovat toteuttaneet kurssin erikseen. Tämä on toki myös mielekäs tapa toteuttaa ohjelmoinnin opetus. Tässä tavassa kuitenkin yhteys matematiikkaan ei välttämättä käy niin vahvasti ilmi kuin ensimmäisellä tavalla toteuttamisessa. Tässä tavassa on

myös se riski, että mikäli kurssia ei toteuteta matematiikan yhteydessä, kurssi järjestetään valinnaisena kurssina. Tällöin ohjelmoinnin opetus ei tavoittane kaikkia oppilaita, jolloin osa oppialaista ei saisi välttämättä ollenkaan ohjelmoinnin opetusta. Kolmannen vaihtoehdon valinneet vastaajat ovatkin luultavasti toteuttaneet ohjelmointia muissa aineissa kuin matematiikassa. Tämä voi tarkoittaa esimerkiksi sitä, että opettaja on toteuttanut ohjelmoinnin opetusta esimerkiksi fysiikassa muttei matematiikassa. Viimeisen vastausvaihtoehdon valinneet vastajat eivät ole toteuttaneet ohjelmoinnin opetusta ollenkaan. Tämä ei tietenkään ole opetussuunnitelman perusteiden tavoite, mutta lienee mielenkiintoista saada tietoa, montako opettajaa on jättänyt ohjelmoinnin opetuksen kokonaan pois.

3.4.3 Kolmas osio: opettajan oma ohjelmointitausta ja kouluttautuminen

Kyselyn viimeisen, eli kolmannen osion aloitti kysymys "Oletko opiskellut itse ohjelmointia ja/tai ohjelmoinnin pedagogiikkaa?". Kysymys oli kyselyn seitsemästoista kysymys. Vastaaaja pystyi vastausvaihtoehdoilla kertomaan, oliko hän opiskellut sekä ohjelmointia että ohjelmoinnin pedagogiikkaa, pelkästään jompaa kumpaa, vai peräti ei kumpaakaan. Tämän kysymyksen avulla saadaan näkemys siitä, miten moni vastaajista on tiettyssä mielessä "pätevä" opettamaan ohjelmointia. Opiskelulla tässä kontekstissa tarkoitetaan sekä yliopistossa opiskelemista, (aiheisiin liittyviin) koulutuksiin osallistumista ja kyseisten aiheiden itseopiskelemista. Tämä kysymys on tämän tutkielman yksi merkittävimmistä, sillä sen avulla voidaan tutkia, miten paljon opettajan ohjelmointikokemus on vaikuttanut esimerkiksi motivaatioon toteuttaa ohjelmointia myös muissa aineissa yhteistyönä (kysymys 12), tai onko kokemuksella vaikutusta tapaan, jolla ohjelmoinnin opetus on toteutettu koulussa (kysymys 16).

Kahdeksantentoista kysymyksenä kysyttiin vastaajalta sitä, kuinka paljon hänellä on ohjelmointikokemusta. Kysymykseen pystyi vastaamaan yhdellä viidestä vaihtoehdosta: "Ei yhtään.", "Olen nähnyt lyhyitä ja yksinkertaisia koodinpätkiä matematiikan oppikirjassa", "Olen silmäillyt materiaaleja netistä, kuten esimerkiksi Code.org tai ohjelmointikurssi.github.io.", "Olen kirjoittanut itse muutaman rivin koodia." ja "Olen harrastanut ohjelmointia vapaa-ajallani.". Kysymyksen tarkoituksena oli selvittää, kuinka paljon opettajilla on kokemusta ohjelmoinnista sekä käyttää tätä kysymystä taustamuuttujana joitakin muita kysymyksiä tutkiessa. Esimerkkinä mainittakoon, että tätä kysymystä voitaisiin käyttää taustamuuttujana tutkiessa, onko vastaaja toteuttanut ohjelmointia muissa aineissa yhteistyönä (kysymys 12). Kysymyksen vastauksista voinee jossain määrin päätellä, miten paljon ohjelmointiharrastuneisuutta vastaajalla on. Samalla kysymyksen tulokset antanevat suuntaa siitä, minkälainen motivaatio opettajilla on ohjelmointia kohtaan.

Kysymyksessä 18 vastausvaihtoehtoja oli siitä syystä paljon, että haluttiin saada selkeä kuva siitä, miten paljon vastaajalta löytyy kokemusta ohjelmoinnin parista. Ensimmäinen vaihtoehto, "Ei yhtään", antaa viitteitä siitä, että vastaajaa ei ole uteliaisuutta saatika kiinnostusta ohjelmointia tai sen opetusta kohtaan. Toinen vaihtoehto, "Olen nähnyt lyhyitä ja yksinkertaisia koodinpätkiä matematiikan oppikirjassa", joka kuvastaa sitä, että vastaajalla on hieman uteliaisuutta ohjelmointia kohtaan, mutta ei juurikaan intoa selvittää mitä muuta ohjelmointi pitää sisällään. Kolmas vastausvaihtoehto, "Olen silmäillyt materiaaleja netistä, kuten esimerkiksi Code.org tai ohjelmointikurssi.github.io.", kuvaa tilannetta, jossa vastaaja on perehtynyt hieman siihen, mitä ohjelmointi pitää sisällään ja miten mahdollisesti sitä voisi toteuttaa. Ajatus neljännen vaihtoehdon, "Olen kirjoittanut itse muutaman rivin koodia.", taustalla oli se, että vastaaja on tosiaan kirjoittanut itse muutaman rivin koodia. Tästä seuraisi se, että hän on itse myös kokeillut mahdollisesti joko kirjasta tai netistä löytämästään materiaalista löytämiään esimerkkejä. Tällöin myös vastaajalle on mahdollisesti myös selvinnyt itse ohjelmoidessaan, kuinka tärkeää syntaksin noudattaminen ohjelmoinnissa todellakin on. Viides ja viimeinen vastausvaihtoehto, "Olen harrastanut ohjelmointia vapaa-ajallani", vastaa tilannetta, jossa vastaajalla on jo selkeästi harrastuneisuutta ohjelmoinnissa. Vastaaja on mahdollisesti kokeillut itse muokata kirjoista tai netistä löytämiään materiaalien koodeja, jolloin hänelle on jäänyt mielikuva siitä, miten monipuolista ohjelmointi on.

Vastausvaihtoehdot olivat tässä kysymyksessä sellaisessa järjestyksessä, jonka uskotaan olevan merkityksellinen vastaajan ohjelmointikokemusta selvitettäessä. Ensimmäinen vaihtoehto pyrki kuvaamaan tilannetta, jossa vastaajalla ei ole yhtään minkäänlaista ohjelmointikokemusta. Toinen ja kolmas vastausvaihtoehto kuvaavat jossain määrin tilannetta, jossa vastaaja tietää pinnallisesti mitä ohjelmointi tarkoittaa, mutta samalla vastaajalla oleva ohjelmointikokemus on hyvin vähäistä. Tämän vaihtoehdon valinneet vastaajat pystyivät seuraamaan jotakin ohjelmointikieltä esimerkiksi, mutta eivät välttämättä pysty itse vielä toteuttamaan vastaavaa ohjelmaa käytännössä. Jos vastaaja valitsi neljännen vastausvaihtoehdon, voidaan hänellä ajatella olevan kohtalainen ohjelmointikokemus. Tähän kategoriaan kuulunevat ne opettajat, jotka osaavat kopioida esimerkkikoodia itse, ja saada siitä toimivan ohjelman aikaan. Mikäli vastaaja valitsi viimeisen vastausvaihtoehdon, voidaan ajatella vastaajalla olevan hyvin ohjelmointikokemusta. Tämän vaihtoehdon valinneiden vastaajien voidaan ajatella omaavan sellaiset ohjelmointitaidot, joilla voi jopa kehittää itse melko yksinkertaisia ohjelmia ja jopa ohjelmointitehtäviä.

Kyselyn viimeinen varsinainen kysymys ennen vapaa sana -osiota oli "Oletko osallistunut koulutuksiin ohjelmoinnin opetukseen liittyen?". Vastausvaihtoehtoja oli kolme: "Olen osallistunut oman kouluni järjestämään koulutukseen.", "Olen osallistunut muun tahon puolesta (esimerkiksi MAOL-päivät) järjestämään koulutukseen." ja "Muu (Mihin?). Kysymyksellä haluttiin ensiksikin selvittää järjestetäänkö vastaajan koulussa koulutuksia ohjelmoinnin opetukseen liittyen ja toisaalta sitä, miten aktiivisesti opettajat

osallistuvat koulutuksiin, jotka liittyvät ohjelmoinnin opetukseen. Matemaattisten Aineiden Opettajien Liitto MAOL ry, joka lienee kansallisesti tunnetuin koulutusten järjestäjä, järjestää koulutuksia matemaattisten aineiden opettajille. Viimeinen vaihtoehto voidaan tulkita hyvin avoimeksi, joten tähän kohtaan on jätetty myös tekstikenttä tarkennusta varten. Esimerkiksi viimeiseen kohtaan on voinut ajatella kuuluvan MOOCit (Massive Open Online Course), eli massiiviset avoimet verkossa järjestettävät kurssit. MOOCien etu piilee siinä, että kaikki kurssimateriaali on vapaasti luettavissa verkossa, joten kaikilla Suomen opettajilla olisi mahdollisuus käydä sama kurssi. Olettaen tietysti, että opettajalla on tietokone käytettävissään sekä toimiva verkkoyhteys. Usein MOOCit ovat täysin ilmaisia ja mahdollistavat kurssin suorittamisen kohtuullisen avoimella aikavälillä omaan tahtiin. Vaihtoehtoista puuttui valitettavasti vaihtoehto "En ole osallistunut koulutuksiin liittyen ohjelmoinnin opetukseen". Tämä on harmillista, mutta tässä tapauksessa vastaaja pystyi esimerkiksi vastaamaan "Muu" ja tarkentaa tekstikenttään "En ole osallistunut".

3.4.4 Vapaa sana

Vapaa sana -osio liitetään yleensä kyselylomakkeiden viimeiseksi kysymykseksi tai kohdaksi. Tämän osion avulla pyydetään vastaajalta yleensä palautetta koko kyselystä sekä muita mahdollisia aiheisiin liittyviä mielteitä, joita vastaajalle on tullut mieleen. On myös tietysti mahdollista, että kyselyyn vastanneelle on tullut opetustyössään vastaan aiheeseen liittyvä ongelmakohta, joka kaipaa kipeästi siihen puuttumista tai mahdollisesti lisätutkimusta aiheesta. Tällöin vastaajat voivat vaikuttaa siihen, miten aihetta tutkitaan lähitulevaisuudessa.

3.5 Tutkimusaineisto

Tätä tutkielmaa varten kerätty tutkimusaineisto koostuu 46:stä matemaattisten aineiden opettajan vastauksesta. Vastaajat valikoitiin kohderyhmään siten, että he opettavat ainakin matematiikkaa yläkoulussa. Koska kyselyssä ei kysytty vastaajan paikkakuntaa tai koulua, missä hän opettaa, ei voida varmuudella tietää, onko vastaajien joukossa kahta (tai useampaa) vastaajaa, jotka olisivat samasta koulusta.

Vastanneet opettajat ovat mahdollisesti hyvinkin eri paikkakunnilta, mutta on myös mahdollista, että jotkut vastaajat voivat olla samasta koulusta. Kyseisten aineenopettajien vastaukset tätä tutkielmaa varten tehtyyn kyselylomakkeeseen [Kyselylomake] (Kyselylomake löytyy kokonaisuudessaan tämän tutkielman liitteistä.).

Luku 4

Tulosten analysointi

Tässä luvussa perehdytään Kruskal-Wallis -testiin sijaluvuille (ranks), eli Kruskal-Wallis H -testiin [Kruskal, W. & Wallis, A.W. (1952)]. Testi tunnetaan myös nimellä yksisuuntainen ANOVA sijaluvuille (one-way ANOVA on ranks). Testi on parametriton menetelmä sille, ovatko otokset peräisin samasta jakaumasta. Sitä käytetään kahden tai useamman saman- tai erikokoisen otoskoon otannoista. Testillä voidaan siis tutkia esimerkiksi useamman joukon kesken, onko niillä samat keskiluvut. Erityisesti testi on hyödyllinen siitä syystä, että parametrittomana menetelmänä, se ei oleta residuaalien (datapisteen poikkeamista regressiosuorasta) normaali-jakautuneisuutta. Lisäksi testi sopii hyvin ordinaaliasteikolliselle mielipidekyselylle, mikä on tässä tutkimuksessa merkittävä osa.

$$H = (N - 1) \frac{\sum_{i=1}^k n_i (\bar{r}_i - \bar{r})^2}{\sum_{i=1}^k \sum_{j=1}^{n_i} (r_{ij} - \bar{r})^2}, \text{ missä}$$

n_i on havaintojen määrä joukossa i ,

r_{ij} on sijaluku (kaikkien havaintojen keskuudessa) havainnon j joukosta i ,

N on havaintojen lukumäärä kaikista joukoista,

$\bar{r}_i = \frac{\sum_{j=1}^{n_i} r_{ij}}{n_i}$ on kaikkien havaintojen sijalukujen keskiarvo joukossa tai ryhmässä i ,

$\bar{r} = \frac{1}{2}(N + 1)$ on kaikkien sijalukujen r_{ij} keskiarvo.

Ensin määritetään nollahypoteesi H_0 , joka on tässä, että sekä kokeneemmat että vähemmän kokeneet opettajat ovat osallistuneet yhtä aktiivisesti koulutuksiin ohjelmoinnin opetukseen liittyen. Vaihtoehtoinen hypoteesi H_1 on se, että näiden ryhmien välillä on eroa.

Käytetään alfan arvoa $\alpha = 0,05$, saadaan siis 95% varmuudella totuudenmukainen tulos. Alfa kuvaa siis merkitsevyystasoa tilastotieteessä, ja on siis todennäköisyys, jolla nollahypoteesi hylätään, kun nollahypoteesi on tosi. Toisin sanoen, alfa kertoo millä todennäköisyydellä tehdään väärä valinta. Voidaan siis ajatella tietäntasosta riskitasoa.

Lasketaan seuraavaksi vapausasteiden määrä. Koska käytössä on neljä ryhmää, on vapausasteluku yksi vähemmän kuin ryhmien lukumäärä. Saadaan vapausasteluvuksi kolme.

Päätössääntöä määritettäessä tarvitaan χ^2 -taulukon [Seppänen, R. ym. (2005)] arvoa löytyvää vapausastelukua 3 ja lukua $\alpha = 0,05$ vastaavaa arvoa, joka on tässä siis 7,815. Päätössääntönä käytetään siis seuraavaa: jos χ^2 on suurempi kuin 7,815, hylätään nollahypoteesi. Muussa tapauksessa jatketaan nollahypoteesiin uskomista. Tämä ei kuitenkaan tarkoita, että nollahypoteesi välttämättä olisi tosi, testin mukaan ei ole mielekästä olla uskomatta nollahypoteesiin.

Tutkitaan kyselyyn osallistuneiden opettajien koulutuksiin osallistumista ja jaetaan opettajat ryhmiin kokemuksen mukaan.

Lasketaan ensin sijaluvut neljässä (kokemuksen mukaan jaetuissa) joukoissa

$$\begin{aligned}\bar{r}_1 &= \frac{1}{13}(1,5 + 1,5 + 4,5 + 4,5 + 4,5 + 4,5 + 7,5 + 7,5 + 11 + 11 + 11 + 11 + 11) = \frac{91}{13} = 7 \\ \bar{r}_2 &= \frac{1}{21}(16,5 + 16,5 + 16,5 + 16,5 + 16,5 + 16,5 + 24,5 + 24,5 + 24,5 + 24,5 + 24,5 + 24,5 + 24,5 + 24,5 + 24,5 + 24,5 + 24,5 + 31,5 + 31,5 + 31,5 + 31,5 + 34) = \frac{504}{21} = 24 \\ \bar{r}_3 &= \frac{1}{10}(35 + 37 + 37 + 37 + 40,5 + 40,5 + 40,5 + 40,5 + 43,5 + 43,5) = \frac{395}{10} = 39,5 \\ \bar{r}_4 &= \frac{1}{2} = \frac{91}{2} = 45,5\end{aligned}$$

ja keskisijaluku (mean rank) kokonaisotoksen (combined sample)

$$\bar{r} = \frac{1}{46}(91 + 504 + 395 + 91) = \frac{1081}{24} = \frac{47}{2} = 23,5.$$

Lasketaan neliöiden joukkosumma (group sum)

$$\begin{aligned}\sum_{j=1}^k n_j(\bar{r}_j - \bar{r})^2 &= 13 * (7 - 23,5)^2 + 21 * (24 - 23,5)^2 + 10 * (39,5 - 23,5)^2 \\ &\quad + 2 * (45,5 - 23,5)^2 \\ &= \frac{14145}{2} \\ &= 7072,5.\end{aligned}$$

ja neliöiden kokonaissumma

$$\begin{aligned}\sum_{i=1}^k \sum_{j=1}^{n_i} (r_{ij} - \bar{r})^2 &= (1,5 - 23,5)^2 + (1,5 - 23,5)^2 + (4,5 - 23,5)^2 + (4,5 - 23,5)^2 + \\ &\quad (4,5 - 23,5)^2 + (4,5 - 23,5)^2 + (7,5 - 23,5)^2 + (7,5 - 23,5)^2 + (11 - 23,5)^2 + (11 - 23,5)^2 + \\ &\quad (11 - 23,5)^2 + (11 - 23,5)^2 + (11 - 23,5)^2 + (16,5 - 23,5)^2 + (16,5 - 23,5)^2 + (16,5 - \\ &\quad 23,5)^2 + (16,5 - 23,5)^2 + (16,5 - 23,5)^2 + (16,5 - 23,5)^2 + (24,5 - 23,5)^2 + (24,5 - \\ &\quad 23,5)^2 + (24,5 - 23,5)^2 + (24,5 - 23,5)^2 + (24,5 - 23,5)^2 + (24,5 - 23,5)^2 + (24,5 - \\ &\quad 23,5)^2 + (24,5 - 23,5)^2 + (24,5 - 23,5)^2 + (24,5 - 23,5)^2 + (31,5 - 23,5)^2 + (31,5 -\end{aligned}$$

$$23,5)^2 + (31,5 - 23,5)^2 + (31,5 - 23,5)^2 + (34 - 23,5)^2 + (35 - 23,5)^2 + (37 - 23,5)^2 + (37 - 23,5)^2 + (37 - 23,5)^2 + (40,5 - 23,5)^2 + (40,5 - 23,5)^2 + (40,5 - 23,5)^2 + (40,5 - 23,5)^2 + (43,5 - 23,5)^2 + (43,5 - 23,5)^2 + (45 - 23,5)^2 + (46 - 23,5)^2 = 7979.$$

Nyt

$$H = (n - 1) \frac{\sum_{j=1}^k n_j (\bar{r}_j - \bar{r})^2}{\sum_{i=1}^k \sum_{j=1}^{n_i} (r_{ij} - \bar{r})^2} = (46 - 1) \frac{7072,5}{7979} = 39,88751723.$$

Koska χ^2 on suurempi kuin 7,815, nollahypoteesi hylätään. Täten siis vaihtoehtoinen hypoteesi pätee testin mukaan, eli todetaan, että ryhmien välillä on eroa. Ero voi johtua juurikin opettajien eri kokemustaustoista. Juuri valmistuneilla tai vähän aikaa jo opettajina toimineilla voi olla enemmän kokemusta ohjelmoinnista ja sen opettamisesta omilta opiskeluajoiltaan. Toisaalta vastavalmistuneilla voi olla enemmän intoa kouluttautua uudehkossa työtehtävässään. On myös mahdollista, että kokeneet opettajat ovat nähneet paljon muutoksia uransa aikana, joten heitä saattaa kyllästyttää jatkuva muutos. Mahdollisia syitä havaittavalle erolla on monia eikä näistä taustalla olevista syistä voida olla varmoja. Äsken tehdyn testin avulla nähdään ainoastaan se, että ryhmien välillä on eroa.

4.1 Kysymysten formulointi

Tässä luvussa käydään läpi kyselylomakkeen kysymysten formulointia, eli sanallista muotoilua. Kysymyksen asettelussa sanallinen muotoilu on äärimmäisen tärkeää tutkimusta tehdessä, jotta tutkitaan oikeata asiaa. Tämän lisäksi pohditaan, onko formulointi onnistunut, eli onko kysymyksillä saatu sentyyppisiä vastauksia, joita haettiin, vai ymmärrettiinkö kysymykset aivan erilailla kuin oli alun perin tarkoitus.

Tutkielma oli tyypiltään kyselytutkimus, jonka otanta oli 46 vastausta. Tätä tutkielmaa varten, näitä tuloksia päätettiin tutkia sekä kvalitatiivisen että kvantitatiivisen tulokinnan menetelmin. Useampaa kysymystä käsiteltiin kvalitatiivisesti samalla kun yhtä kysymystä tulkittiin kvantitatiivisesti.

Kyselyn kysymysten formulointi lienee useimmiten haastavin osa kyselytutkimusta tehdessä. Usein riskinä on se, että tarpeeksi kattavassa otannassa löytyy ihmisiä, jotka saattavat tulkita kysymyksen väärin. Tämän tutkielman osalta muotoiluun kiinnitettiin erityistä huomiota. Pääosin kyselyn kysymykset olivat muotoilultaan juuri sellaisia, kuten oli alun perin ajateltu, sillä niillä saatiin juuri sen suuntaisia vastauksia kuin mitä odotettiin. Ainoa kysymys, jonka voi kyselyssä tulkita helpohkosti eri lailla kuin mitä tarkoitettiin, on kysymys 15. *Millä ohjelmointikielellä olet toteuttanut ohjelmointia? (Valitse vähintään yksivaihtoehto, voit valita useita).* Kysymyksellä tarkoitettiin tarkemmin sanoen sitä, että millä ohjelmointikielellä vastaaja on toteuttanut ohjelmoinnin opetusta,

eikä se käy kysymyksestä selväksi suoraan. Tästä syystä kysymyksen vastaukset päätettiin jättää huomioimatta. Tarkoituksenmukaisempi muotoilu olisi kysymykselle olisi ollut *Millä ohjelmointikielellä olet toteuttanut ohjelmoinnin opetusta? (Valitse vähintään yksi-vaihtoehto, voit valita useita).*

4.2 Kyselylomakkeen tulokset ja huomiot

Ohjelmointia voidaan pitää kohtuullisen uutena ilmiönä nykypäivän koulumaailmassa. Vaikkakin ohjelmointi on ollut joidenkin koulujen valinnaisena kurssina, ei sitä ole sisällytetty laajemmin perinteisten oppiaineiden sisään. Yksi asia, joka lienee korostavan ohjelmoinnin opetuksen uutuutta suomalaisessa koulutusjärjestelmässä, on se, että tutkimuksessa havaittiin vastaajien käyttävän mitä erinäisimpiä termejä. Ohjelmoinnillinen ajattelu, ohjelmallinen ajattelu, algoritmiajattelu, algoritmien ajattelu, tämä viitannee siihen, että joko vastaajat eivät tiedä tarkkaan mistä termissä on kyse, tai vaihtoehtoisesti ilmiön uutuudesta koulumaailmassa. Suomalaisten matematiikan opettajien keskuuteen ei ole vakiintunut osuvaa termiä taidoille, joita ohjelmoinnin avulla halutaan opettaa. Seuraavaksi avataan sanallisesti, miten kyselylomake rakentui ja mikä oli kyselylomakkeen tarkoitus tässä tutkielmassa.

Kysymykset, joihin tämä tutkielma pyrki löytämään vastauksen, käsitellään seuraavaksi kappaleittain. Tutkielman avulla pyrittiin tutkimaan, miten opettajat ovat toteuttaneet ohjelmoinnin opetusta matematiikan opetuksessa, ja tästä erityisesti kyselyn kysymykset kuten kysymys 16 (Onko koulussanne toteutettu ohjelmoinnin opetus matematiikassa vai pidetäänkö kurssi erikseen?) ja kysymys 7 (Miten olet toteuttanut ohjelmointia matematiikan opetuksessa?) olivat erityisen mielenkiinnon kohteena. Lisäksi kyselytutkimuksen avulla saatiin tietoa siitä, käyttävätkö opettajat keskenään samoja materiaaleja, onko opettajilla käytössä tarpeeksi resursseja (kuten aikaa, tietokoneita ja oppimateriaaleja) sekä ovatko opettajat toteuttaneet ohjelmointia muissa oppiaineissa (esimerkiksi fysiikassa) yhteistyönä. Kyselytutkimuksen vastausten avulla voitaneen arvioida, miten opettajan oma asenne ohjelmointia ja sen opetusta kohtaan on vaikuttanut tapoihin opettaa ohjelmointia matematiikan opetuksessa. Kyselytutkimuksen tuloksia analysoidessa havaittiin trendi opettajien ohjelmoinnin opetuksen ajoittamisen painottamisessa tiettyihin ajanjaksoihin. Huomautettakoon, että kyselytutkimus ehdittiin valitettavasti toteuttaa ennen kuin uusi opetussuunnitelma ehti vaikuttaa yhdeksännen luokan opetusryhmiin, joten niillä vastaajilla, joilla oli ollut vain yhdeksännen luokan opetusryhmiä, ei ollut ollut mahdollisuutta opettaa ohjelmointia uuden opetussuunnitelman mukaisesti.

4.2.1 Opettajien motivaatio ohjelmoinnin opetusta kohtaan

Seuraavaksi pohditaan, miten opettajien oma asenne ohjelmoinnin opetusta kohtaan (kysymys 4) ja kuinka paljon opettajien ohjelmointikokemus vaikuttavat toisiinsa (kysymys 18). Pohdinnasta mielenkiintoisen tekee se, että aiemmin matematiikan opettajien ei ollut pakko opettaa ohjelmointia. Nykyään kaikkien matematiikan opettajien tulisi opettaa ohjelmointia matematiikan opetuksen yhteydessä. Ymmärrettävästi tämä voi harmittaa osaa opettajista, sillä heidän täytyy osata jotain, mitä ei ole aiemmin opiskellut, eikä välttämättä koe mielenkiintoiseksi. Koska jaettaessa vastaajat luokkiin ohjelmointikokemuksen määrän perusteella, kysymysten tulokset ristiintaulukoitiin [Ristiintaulukointi]. Muuten tulokset eivät olisi olleet vertailukelpoisia keskenään.

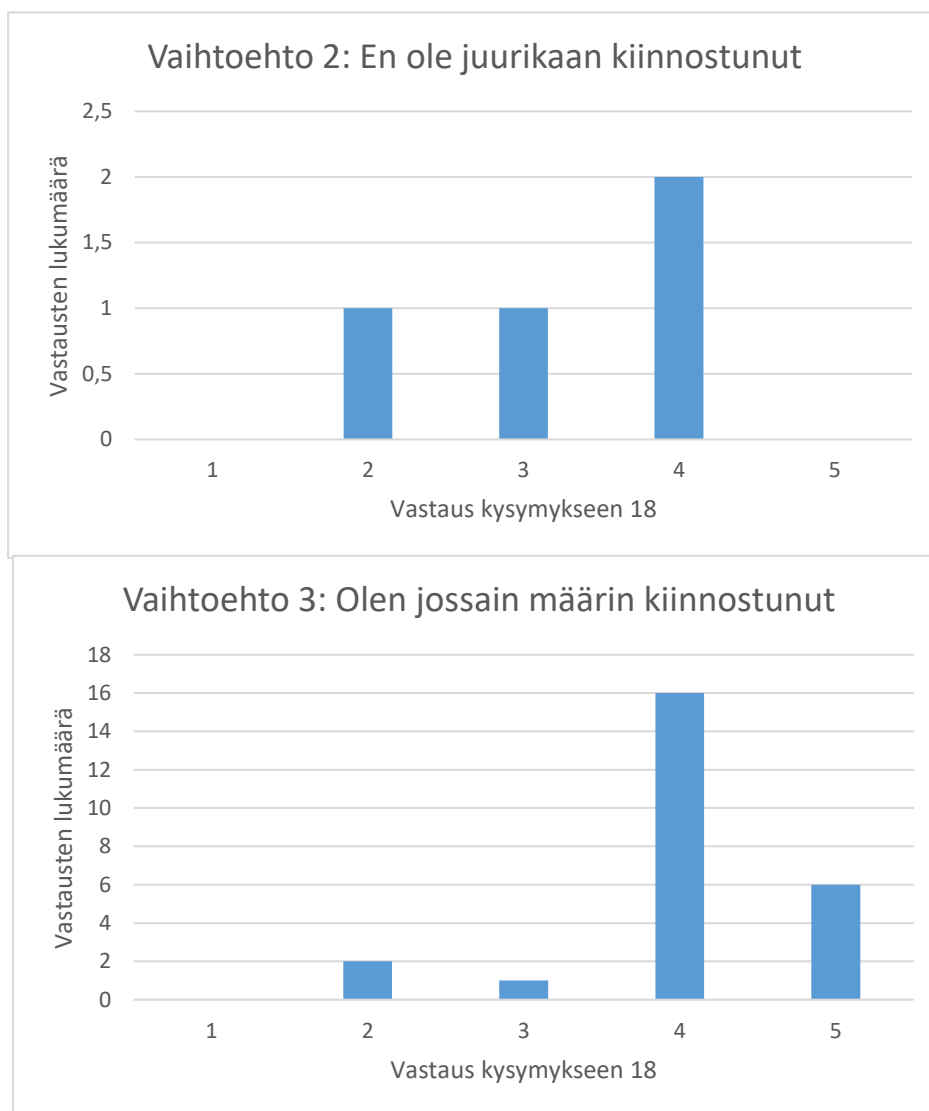
Tutkielmassa haluttiin selvittää, kuinka opettajien oma asenne ohjelmoinnin opetusta kohtaan korreloi opettajan ohjelmointikokemusten määrän kanssa. Kyselyn tuloksia analysoitaessa havaittiin, että ne opettajat, jotka ovat erittäin kiinnostuneita (13,0%) (kysymys 4, vaihtoehto 5), ovat joko kirjoittaneet itse muutaman rivin koodia (erittäin kiinnostuneista 33,3%) tai harrastaneet ohjelmointia vapaa-ajallaan (erittäin kiinnostuneista 66,7%). Ohjelmoinnin opettamisesta hyvin kiinnostuneet opettajat (23,9%) (kysymys 4, vaihtoehto 4) olivat vastanneet joko silmälleen materiaaleja netistä (hyvin kiinnostuneista 18,2%), kirjoittanut itse muutaman rivin koodia (hyvin kiinnostuneista 45,5%) tai harrastanut ohjelmointia vapaa-ajallaan (hyvin kiinnostuneista 36,4%). Jossain määrin ohjelmoinnin opetusta kohtaan kiinnostuneita (54,3%) (kysymys 4, vaihtoehto 3) opettajia oli vastaajista eniten. Nämä opettajat olivat joko nähneet lyhyitä ja yksinkertaisia koodinpätkiä matematiikan oppikirjoissa (jossain määrin kiinnostuneista 8,0%), silmälleen materiaaleja netistä (jossain määrin kiinnostuneista 4,0%), kirjoittanut itse muutaman rivin koodia (jossain määrin kiinnostuneista 64,0%) tai harrastanut ohjelmointia vapaa-ajallaan (jossain määrin kiinnostuneista 24,0%). Vastaajat, jotka eivät olleet juurikaan kiinnostuneita ohjelmoinnin opettamisesta (8,7%)(kysymys 4, vaihtoehto 2) olivat vastanneet joko nähneensä lyhyitä ja yksinkertaisia koodinpätkiä matematiikan oppikirjassa (ei juurikaan kiinnostuneista 25,0%), silmälleen materiaaleja netistä (ei juurikaan kiinnostuneista 25,0%) tai kirjoittaneensa itse muutaman rivin koodia (ei juurikaan kiinnostuneista 50,0%). Kukaan kyselyyn osallistuneista opettajista ei ollut vastannut, että hän ei olisi ollut yhtään kiinnostunut ohjelmoinnin opetuksesta (0,0%)(kysymys 4, vaihtoehto 1).

Kyselyn tuloksista voidaan havaita selvää korrelointia luokkien ja vastausten välillä. Ohjelmoinnista vähemmän kiinnostuneet opettajat eivät harrasta ohjelmointia vapaa-ajallaan, sen sijaan jossain määrin kiinnostuneista vastaajista peräti 24,0% kertoo harrastaneensa ohjelmointia vapaa-ajallaan. Ohjelmoinnin opetusta kohtaan erittäin kiinnostuneesti suhteutuneista opettajista ohjelmointia vapaa-ajallaan oli harrastanut jopa 66,7%.

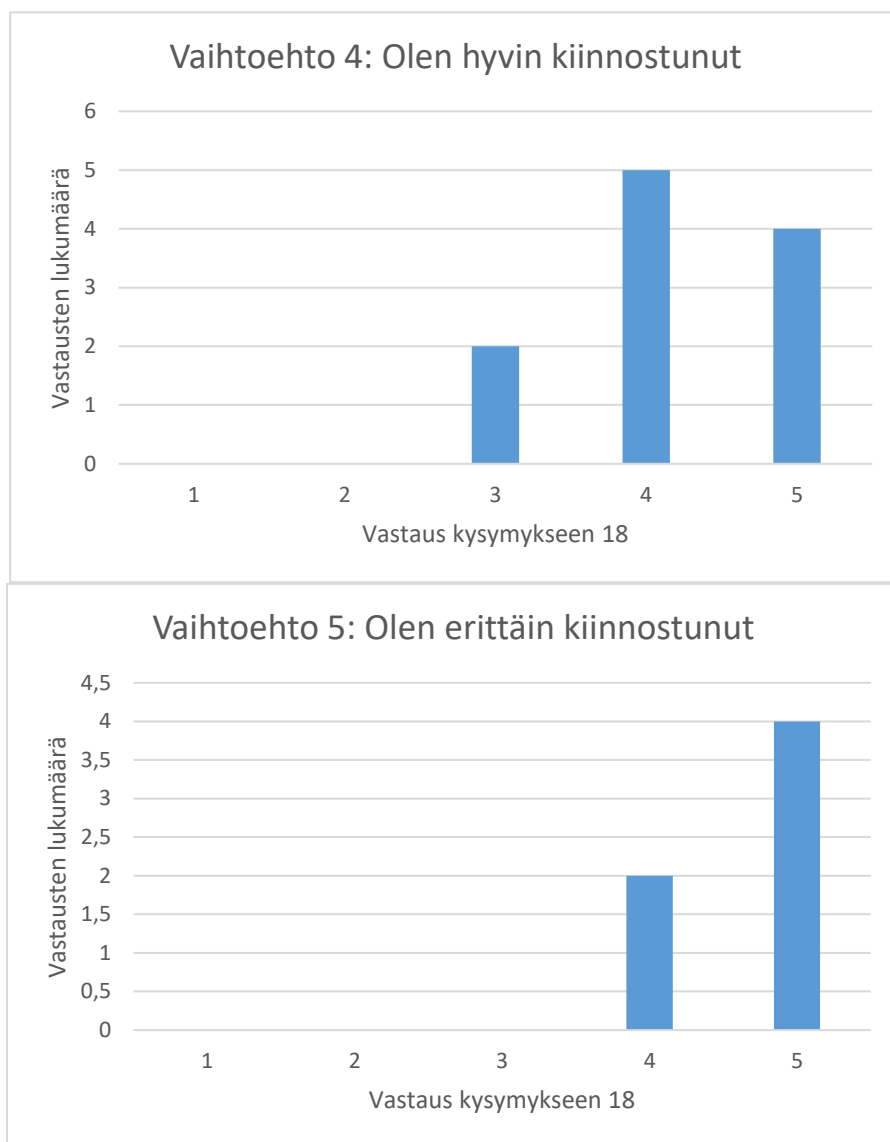
Alla olevissa kaavioissa kysymyksen 18 (Kuinka paljon sinulla on ohjelmointikoke-

musta?) vastausvaihtoehdot ovat seuraavat: Ei yhtään (vastausvaihtoehto 1), Olen nähnyt lyhyitä ja yksinkertaisia koodinpätkiä matematiikan oppikirjassa (vastausvaihtoehto 2), Olen silmäillyt materiaaleja netistä, kuten esimerkiksi Code.org tai ohjelmointikurssi.github.io (vastausvaihtoehto 3), Olen kirjoittanut itse muutaman rivin koodia (vastausvaihtoehto 4) ja Olen harrastanut ohjelmointia vapaa-ajallani (vastausvaihtoehto 5).

Kuva 4.1: Kysymyksen 4 vastausvaihtoehdot 2 (ylempi kaavio) ja 3 (alempi kaavio).



Kuva 4.2: Kysymyksen 4 vastausvaihtoehdot 4 (ylempi kaavio) ja 5 (alempi kaavio).



4.2.2 Ohjelmoinnin opetus matematiikassa vai erikseen valinnaisena kurssina

Yksi tämän tutkielman mielenkiinnonkohteista oli se, onko ohjelmointia opetettu matematiikan oppitunneilla vai toteutetaanko ohjelmoinnin opetusta matematiikasta erillisellä kurssilla. Kyselylomakkeeseen vastanneista opettajista 33 vastaajaa oli kertonut toteuttavansa ohjelmoinnin opetuksen matematiikassa. Kolme vastaajaa oli valinnut vaihtoehdon "Kurssi pidetään erikseen.". Yksi vastaaja oli ilmoittanut, että hänen koulussaan ohjelmoinnin opetusta ei tapahtuisi. Sen sijaan loput yhdeksän vastaajaa oli valinnut vaihtoehdon "Muu". Viimeisen vaihtoehdon yhteydessä pyydettiin tarkentamaan tapaa mikä on tämä "Muu" tapa. Tarkennuksista löytyi seuraavanlaisia vastauksia: "Perusopetus matematiikassa, täydentävää valinnaisessa media/tietotekniikassa tai ohjelmoinnissa, mutta tietystikään kaikki oppilaat eivät ole valinneet niitä", "Meillä on ollut vanhassa opsissa Teknologia-niminen valinnaiskurssi, jossa on ollut ohjelmointiosio. Uudessa opsissa ohjelmointi-valinnaisaine on alaluokkien puolella, mutta matematiikan tunneillakin opetetaan ohjelmoinnillista ajattelua eri vuosiluokilla.", "OPSin mukainen opetus on matematiikassa ja käsitöissä, meillä on erikseen valinnainen AT, jossa ohjelmointia.", "matikassa on ollut myös lyhytkurssi ohjelmoinnista valinnaisena", "Matematiikan opettajat voivat halutessaan pitää matematiikan tunneilla. ATK-opettaja pitää ohjelmoinnin opetusta myös atk-kursseilla.", "Kaikille matikassa, valinnaisena tietotekniikan lyhytkurssi.", "Sekä että.", "Matematiikan opetuksessa ja myös tietotekniikassa." ja "sekä matematiikassa että erikseen". Kuten vastauksista huomataan, osa vastaajista tähdensi, että näiden kahden vaihtoehdon lisäksi löytyy myös kolmas vaihtoehto, sillä nämä vaihtoehdot eivät tietenkään ole toisensa poissulkevia. Näitä vaihtoehtoja yhdistää siis yksi piirre, ohjelmoinnin opettamisen monipuolisuus. Ohjelmoinnin opettamista ei ole välttämätöntä rajoittaa ainoastaan yhden oppiaineen vastuulle, kuten uusimmassa opetussuunnitelmasakin [OPS 2014] sanotaan.

Äskeisen perusteella voidaan todeta tämän tutkielman valossa, että enemmistössä ovat ne opettajat, joiden kouluissa ohjelmointia opetetaan ainakin matematiikassa. Ainoastaan kolme opettajaa neljästäkymmenestäkuudesta opettajasta oli ilmoittanut, että heidän kouluissaan matematiikassa ohjelmoinnin opetusta ei tapahtuisi.

Niissä kouluissa, joissa ohjelmoinnin opetus oltiin toteutettu omana oppiaineenaan, ilmenee useita variaatioita toteutuksissa. Eräässä koulussa, jossa käytössä oli ohjelmoinnin opetus omana oppiaineenaan, oli oppiaineen tai kurssin nimi "Media ja tietotekniikka". Kyseisen kurssin aiheena kahdeksannella vuosiluokalla aiheena oli robotiikka. Samassa koulussa vanhassa opetussuunnitelmassa oli "Teknologia" -niminen valinnaiskurssi, kun uudessa opetussuunnitelmassa koulussa oli käytössä "ohjelmointi" -valinnaisaine alaluokkien puolella. Sama vastaaja kertoi, että hänen opettamillaan matematiikan oppitunneilla opetetaan ohjelmoinnillista ajattelua eri vuosiluokilla.

4.2.3 Tapoja toteuttaa ohjelmoinnin opetusta yläkoulussa

Toinen tämän tutkielman mielenkiinnonkohteista oli se, kuinka ohjelmoinnin opetusta on toteutettu yläkoulussa. Kyselytutkimuksesta kävi ilmi, että vaikka tapoja on monia, niin joitakin yhteisiä piirteitä kuitenkin löytyy. Ensinnäkin ohjelmoinnin opetusta on toteutettu yhteistyössä muiden oppiaineiden kanssa. Näistä kyselyssä ilmenneitä esimerkkejä ovat tietotekniikka, fysiikka, kemia, biologia, käsityöt, valinnainen luonnontiede sekä erilaiset monialaiset oppimiskokonaisuudet. Vaikka opetussuunnitelmasta äsken mainituista oppiaineista maininta löytyy lähinnä käsitöiden kohdalla, lienevät monialaiset oppimiskokonaisuudet olevan enemmänkin rikkaus kuin rasite. Tässä tutkielmassa ei valitettavasti kysytty tarkentavasti, kuinka ohjelmoinnin opetusta oppiaineiden välillä on toteutettu. Tästä voisikin saada kenties kokonaan uuden ja erillisen tutkielman.

4.2.4 Tapoja toteuttaa ohjelmoinnin opetusta matematiikan opetuksessa

Tässä alaluvussa poraudutaan tarkemmin tapoihin toteuttaa ohjelmoinnin opetusta matematiikan opetuksessa. Ikävä kyllä kyselytutkimus sattui hieman hankalaan vaiheeseen, sillä kuten yksi vastaajakin huomautti: kaikilla opettajilla ei välttämättä ollut ollut uuden opetussuunnitelman mukaista opetusryhmää. Tämä tilanne saattoi tapahtua esimerkiksi siinä tilanteessa, että opettajalla oli ainoastaan yhdeksännen vuosiluokan oppilaita ryhmässään kyseisenä lukuvuonna. Tästä syystä tällä kyselytutkimuksella ei voitu tavoittaa aivan kaikkia potentiaalisia vastaajia.

Kyselyn tuloksia analysoitaessa havaittiin trendi, jonka mukaan moni opettaja opettaisi ohjelmointia vain tiettyinä ajanjaksoina sen sijaan, että olisi opettanut toistuvasti koko lukuvuoden. Esimerkiksi jotkut opettajat olivat valinneet opettaa ohjelmointia vasta kokeiden jälkeen. Tämä vaihtoehto lienee perusteltua, sillä toistaiseksi kokeissa ei taida juurikaan esiintyä ohjelmointitehtäviä, jolloin on hyvin merkittävää opettaa ensisijaisesti ne asiat, joita kokeissa kysytään. Joissakin kouluissa käytössä oli ohjelmointiviikko. Osassa näist kouluista kyseinen ohjelmointiviikko oli ainoastaan jompanakumpana lukukaute-na, joissakin sen sijaan molempina lukukausina. Eräässä koulussa oltiin valittu toteuttaa ohjelmoinnin opetus joko ennen joulua tai vaihtoehtoisesti ennen hiihtolomaa. Näissä molemmissa tavoissa oli kuitenkin yhteistä se, että ohjelmoinnin opetus toteutettiin vasta kokeiden jälkeen. Jotkut opettajat olivat valinneet toteuttaa ohjelmoinnin opetuksen valikoituina viikkoina, yleisimmin joko viikko tai pari, niissä tapauksissa kun muu opetus oli edennyt sopivaan pisteeseen. Vastauksista käy myös ilmi, että joissakin kouluissa käytetään ohjelmoinnin opetusta täyteohjelmanä, eli silloin, kun kirjan omat aiheet on käyty läpi. Lähimmäksi opetussuunnitelman tavoitteita pääsevät ne opettajat, jotka ovat toteuttaneet esimerkiksi ohjelmoinnin opetuksen kaksoistunnin jälkimmäisellä tunnilla, tai

kun perinteinen tunti on korvattu ohjelmoinnilla.

4.2.5 Onko opettajilla tarpeeksi resursseja ohjelmoinnin opettamiseen

Tämä kysymys jaettiin kolmeen merkittävästi erilliseen kategoriaan: aika, tietokoneet ja oppimateriaalit. Kirjoittaja koki nämä kolme kategoriaa merkittävimmiten osiksi mitä tulee opettajien halukkuuteen opettaa ohjelmointia. Mikäli jotain näistä opettajalla ei ole riittävästi, saattaa opettajan motivaatio opettaa ohjelmointia laskea merkittävästikin. Tämän kysymyksen avulla saadaan arvokasta tietoa siitä, ovatko opettajat eriarvoisessa asemassa esimerkiksi vähäisten tai peräti puuttuvien resurssien takia joillakin alueilla.

Tietokoneet (kyselylomakkeen kysymys 10, osa a.)

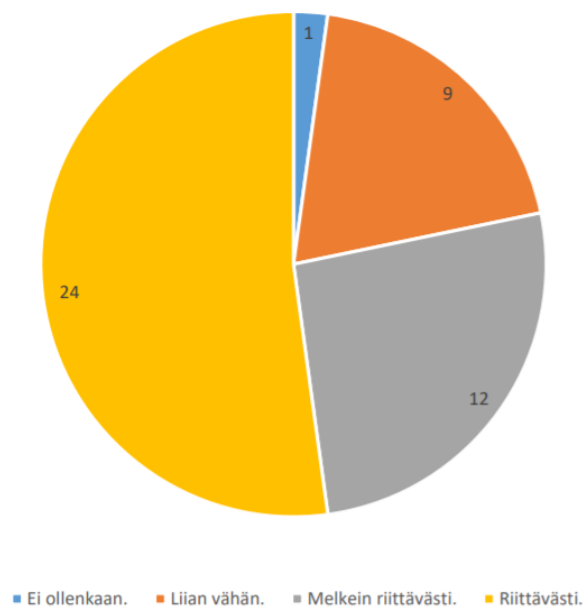
Ohjelmoinnin opetus ilman tietokoneita ja sopivaa ohjelmointiympäristöä on kuin kuvaamataidon opettamista ilman paperia ja kyniä - "opiskelu" tapahtuisi ainoastaan ajatuksen tasolla. Kuitenkin käytäntöä voitaneen pitää vähintäänkin yhtä merkittävänä osana ohjelmoinnin opetusta kuin itse ohjelman toiminnan pohtimista ajatuksen tasolla. Tästä syystä riittävän syvällisen ymmärtämisen tason saavuttamiseksi on ehdottoman tärkeää saada harjoitella käytännössä ohjelman laatimista ja kirjoittamista tietokoneella. Tällöin oppilaille voisi jäädä parempi muistijälki, sillä opetuksessa olisi käytetty myös visuaalisuutta.

Opetussuunnitelman tavoitteita ohjelmoinnin opetuksen toteutumisesta voidaan pitää hyvin haastavina, ellei mahdottomina, mikäli kaikilla oppilailla ei ole mahdollisuutta ohjelmoida itse. Vaikkakin esimerkiksi parin kanssa ohjelmoimisesta voi olla myös merkittävää hyötyä, jolloin riittäisi, että joka toiselle oppilaalle olisi oma tietokone, on kyseenalaisista, että yhden tietokoneen pitäisi riittää kuudelle oppilaalle. Kyseinen tilanne on siis todellisuutta erään vastaajan koulussa. Ei liene mielekästä olettaa, että vastaavanlainen tilanne toteuttaa tarkoitustaan. Yhdeksän vastaajaa oli kertonut, että heidän kouluissaan oli ollut käytössä liian vähän tietokoneita ohjelmoinnin opetukseen. Vastanneista yksi oli ilmoittanut, että vastaajan koulussa ei ollut ollut yhtään tietokonetta käytössä ohjelmoinnin opetukseen. Molemmat tilanteet ovat erittäin haasteellisia ja ikäviä tilanteita, mikäli ohjelmointia aikoo toteuttaa niin, kuin opetussuunnitelmassa [OPS 2014] kuvataan.

Kuitenkin enemmistöllä vastanneista on ollut koulussaan riittävästi tai melkein riittävästi tietokoneita käytössä ohjelmoinnin opetuksessa. Arvailujen varaan kuitenkin jää, mitä vastausvaihtoehto "Melkein riittävästi" kullekin vastaajalle tarkoittaa. Vaihtoehdolla voidaan tarkoittaa esimerkiksi tilannetta, että joillakin oppitunneilla aivan kaikille ei riitä tietokonetta. Toinen tulkinta voi olla esimerkiksi sellainen, että opettaja haluaisi toteuttaa ryhmänsä kanssa hieman useammin ohjelmointia tunneilla tietokoneilla, kuin siihen

on mahdollisuutta. Edellinen tilanne voi olla seuraus siitä, että koulussa tietokoneita kaikille ryhmille ei riitä samanaikaisesti. Lohduttavaa on kuitenkin se, että vastanneista yli puolet (24 vastaajaa) kertoi, että koulussa oli ollut tietokoneita riittävästi ohjelmoinnin opetusta toteuttaessa.

Kuinka paljon sinulla on käytössäsi seuraavia resursseja:
a. Tietokoneita



Kuva 4.3: Kyselyyn osallistuneiden vastauksia siitä, kuinka he ovat työssään kokeneet resurssien riittävyyden tietokoneiden osalta.

Oppimateriaalit (kyselylomakkeen kysymys 10, osa b.)

Ilman hyviä oppimateriaaleja opettajien vastuulle voi jäädä melkoinen taakka keksiä tehtäviä tai opettaa ohjelmointiympäristön käyttöä, mikä voi opettajalle itselleenkin olla vierasta. Yksitoista vastaajaa oli kertonut vastauksissaan, että he eivät ole opiskelleet ohjelmointia taikka ohjelmoinnin pedagogiikkaa. Toisaalta vain yhdeksän vastaajaa oli maininnut opiskelleensa sekä ohjelmointia että ohjelmoinnin pedagogiikkaa. Nämä yhdeksän ovat kuitenkin vähemmistössä tämän kyselytutkimuksen otannassa. Ei siis lie- ne mielekästä olettaa, että kaikki opettajat haluaisivat tai edes kykenisivät kehittämään omia oppimateriaaleja ohjelmoinnin opetukseen. Mielekkäämpää on olettaa, että opetta- jat haluavat käyttää jo tehtyjä oppimateriaaleja ohjelmoinnin opetukseen. Opettajien on- neksi näitä löytyy internetistä runsain mitoin [Studio.code.org](https://studio.code.org) [code.org], tie.koodariksi.fi [[Tie.koodariksi.fi](https://tie.koodariksi.fi)] ja ohjelmointikurssi.github.io [[Ohjelmointikurssi.github.io](https://ohjelmointikurssi.github.io)], kunhan tie- tää mistä etsiä. Äsken mainittujen sivustojen tehtävät saa suomenkielellä ja ovat har- joittavat yleisiä ohjelmoinnissa tarvittavia taitoja. Ensimmäisenä mainittu sopii hyvin myös alakouluihin, toisena mainittu sopii erinomaisesti johdatukseksi ohjelmoinnin al- keisiin Python-ohjelmointikielellä ja kolmas on kattava perehdytys ohjelmoinnin alkeisiin JavaScript-kielellä. Äsken mainitut sivustot ovat tietysti vain yksittäisiä esimerkkejä sii- tä, mitä oppimateriaaleja ohjelmoinnin opetukseen liittyen internetistä löytyy.

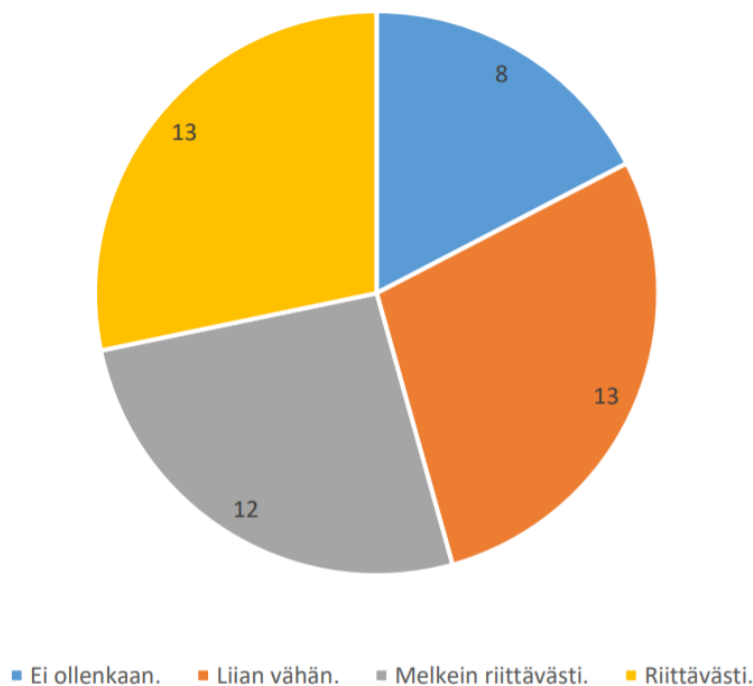
Kysymykseen kymmenen peräti kahdeksan vastaajaa oli vastannut oppimateriaalien osalta, että heidän käytössään ei olisi ollut oppimateriaaleja ohjelmoinnin opetusta var- ten. Tässäkään kysymyksessä vastauksesta ei suoraan käy ilmi, mistä syystä vastaajat ovat valinneet vaihtoehdon "Ei ollenkaan". Osa vastaajista on voinut tarkoittaa vastauk- sensa tulkittavaksi niin, että koulussa käytettävässä kurssin oppikirjassa ei ole yhtään ohjelmointiin liittyviä tehtäviä. Tämä lienee totta joidenkin yläkoulun matematiikan op- pikirjojen osalta. Toinen mahdollinen tulkinta voi kuitenkin olla se, että aiemman lisäksi opettaja ei ole löytänyt internetistä sopivaa sivustoa, jolla ohjelmointia voitaisiin harjoi- tella. Kolmetoista vastaajaa kertoi, että käytössä olevista resursseista ohjelmoinnin ope- tukseen liittyviä oppimateriaaleja olisi ollut käytettävissä "Liian vähän". Näin vastanneet opettajat ovat luultavasti kokeneet, että oppikirjan ohjelmointiin liittyvät tehtävät eivät ole riittäneet kattavaan opetukseen. Samat opettajat ovat mahdollisesti kokeneet että in- ternetistä löytyvät oppimateriaalit eivät ole vastanneet tarpeeksi sitä, mitä heidän tulisi opettaa ohjelmoinnista.

Yli puolet (25 vastaajaa) vastaajista olivat vastanneet kysymykseen myönteisemmin. Kaksitoista vastaajaa olivat ilmoittaneet, että heillä olisi ollut käytössään "Melkein riit- tävästi" oppimateriaaleja ohjelmoinnin opetusta toteuttaessa. Loput vastanneista olivat vastanneet, että käytössä olisi ollut riittävästi oppimateriaaleja. Näissä kahdessa vastaus- vaihtoehdoryhmässä lienee yhteistä se, että merkittävä osa ohjelmoinnin opettamiseen liittyvistä oppimateriaaleista on osattu etsiä myös oppikirjan ulkopuolelta. On mukavaa

huomata, että vastaajat ovat osanneet etsiä materiaaleja oikeista paikoista, mikä kertoo kiinnostuksesta ja oma-aloitteisuudesta kehittää omaa opetustaan. Vaihtoehtoinen tulkinta materiaalien riittävyydelle on se, että opettajalle on riittänyt oppikirjan materiaali ohjelmoinnin opetuksen toteuttamiseen.

Kuinka paljon sinulla on käytössäsi seuraavia resursseja:

b. Oppimateriaaleja



Kuva 4.4: Kyselyyn osallistuneiden vastauksia siitä, kuinka he ovat työssään kokeneet resurssien riittävyyden oppimateriaalien osalta

Aika (kyselylomakkeen kysymys 10, osa c.)

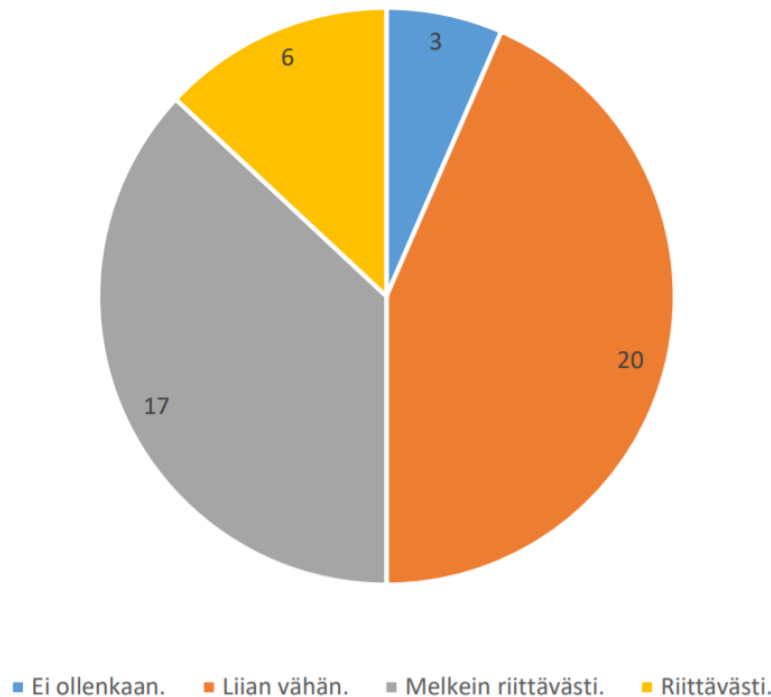
Siinä missä matematiikka vaatii aikaa erinäisten käsitteiden syvälliseen ymmärtämiseen ja teorian soveltamiseen käytännössä, vaatii myös ohjelmointinnin harjoitustehtävien tekeminen ajatustyötä. Erityisesti silloin, kun kyseessä on haastavat ongelmatehtävät. Mikäli oppilailla ei ole riittävästi aikaa esittää kysymyksiä ja pohtia tehtävän ratkaisua itse, ei voida olettaa, että todellista oppimista tapahtuisi. Myöskään ei voida olettaa, että oppilailla olisi riittävästi aikaa pohtia tehtäviä, jos opettajille ei anneta riittävästi aikaa opettaa uutta asiaa. Haastavan opetussuunnitelman tavoitteesta tekee se, että mikäli paikallisten opetussuunnitelmien tuntijaossa matematiikassa ei ole tapahtunut muutosta. Erityisesti opettajien voi olla hankalaa opettaa perinteisiä matematiikan kokonaisuuksia ja tämän lisäksi ohjelmointiin liittyviä kokonaisuuksia kattavasti ja opetussuunnitelman tavoitteiden mukaisesti matematiikan opetuksen rinnalla.

Kysymyksen viimeisessä osassa kysyttiin opettajien käytössä olleiden resurssien osalta aikaa. Mielenkiinnon kohteena oli kysymyksessä se, kuinka paljon heillä on ollut käytössä aikaa ohjelmoinnin opetuksen toteuttamiseen. Vastanneista peräti kolme opettajaa oli ilmoittanut, että heillä ei ollut ollut käytössä ollenkaan aikaa ohjelmoinnin opetuksen toteuttamiseen. Toisen vaihtoehdon "Liian vähän" oli valinnut jopa kaksikymmentä opettajaa. Nämä tulokset ovat huolestuttavia, sillä mikäli opettajat eivät ole kokeneet saaneensa tarpeeksi aikaa opettaa ohjelmointia, ei luultavasti ole saavutettu opetussuunnitelman tavoitteita. Mitä luultavammin oppilaillekaan tuskin on jäänyt vahvaa näkemystä siitä, mitä varten ohjelmointia voisi olla mielekästä opiskella.

Mielenkiintoista kyllä, puolet vastanneista (23 vastaajaa) oli ilmoittanut, että heillä olisi joko "Melkein riittävästi" tai "Riittävästi" aikaa ohjelmoinnin opetukseen. Kahdeksantoista vastaajaa oli vastannut, että heillä olisi käytössään "Melkein riittävästi" aikaa toteuttaessaan ohjelmoinnin opetusta. Tämä on lohduttavaa kuulla, sillä riittävä aika uuden asian opetteluun ja opettamiseen on erittäin tärkeää. Kuitenkin jopa kuusi vastaajaa oli kertonut, että heillä olisi ollut käytössään "Riittävästi" aikaa ohjelmoinnin opettamiseen. Kahden jälkimmäisen vastausvaihtoehdon vastanneet opettajat lienevät saavuttaneen ainakin lähes kokonaan tarpeelliset laajuudet ohjelmoinnin sisällöistä.

Nyt kun kolme kirjoittajan mielestä tärkeintä resurssia ohjelmoinnin opetukseen liittyen on käyty läpi, voidaan todeta, että tämän kyselytutkimuksen valossa resurssit eivät ole jakautuneet koulujen välillä tasapuolisesti. Kyselyyn vastanneet opettajat ovat täten hyvin eriarvoisessa asemassa. Erityisesti, mikäli opettajilla ei ole käytössään tarpeeksi resursseja, saattaa halukkuus opettaa ohjelmointia laskea merkittävästi. Opettaja saattaa tällöin jopa kyseenalaistaa opetussuunnitelman tavoitteen opettaa kaikille ohjelmointia. Seuraavaksi käydään läpi kyselyyn vastanneiden opettajien mietteitä siitä, tarvitsevatko kaikki oppilaat ohjelmoinnin opetusta.

Kuinka paljon sinulla on käytössäsi seuraavia resursseja:
c. Aikaa



Kuva 4.5: Kyselyyn osallistuneiden vastauksia siitä, kuinka he ovat työssään kokeneet resurssien riittävyyden ajan osalta

4.2.6 Tarvitsevatko kaikki oppilaat ohjelmoinnin opetusta

Tämä kysymys lienee jakanut eniten mielipiteitä kaikista kyselylomakkeen kysymyksistä. Yksi perustelu mielipiteiden jakautumiselle lienee se, että itse kysymys on muotoilultaan jopa hieman utopistinen. Toisaalta ajatus siitä, että kaikki oppilaat ja täten myös kaikki ihmiset tarvitsisivat jotain tiettyä taitoa, voi monen mielestä olla liioittelua. Saman kysymyksen voisi esittää melkein mistä tahansa muustakin oppiaineesta, kuten esimerkiksi liikunnasta, englannin kielestä, ruotsin kielestä tai vaikkapa matematiikasta. Kuitenkin perustelu opettamisen puolesta voisi olla seuraavanlainen: Vaikka ihminen pärjää arkipäiväisessä elämässään hyvin ilman tiettyjä taitoja, ei se tarkoita sitä, etteikö näiden taitojen omaamisesta olisi jotain merkittävää hyötyä yksilölle. Siitä huolimatta, että kaikista ei tule ammattilaisia ohjelmistoalalle, voivat ohjelmointitaidot opettaa merkittävästi tärkeitä

taitoja, joita voi tarvita monissa muissakin ammateissa. Osa kyselyyn vastanneista koki kysymyksen selvästi kirjaimellisesti, mutta oli vastaajissa myös niitäkin, jotka ottivat hieman vapaamman tulkinnan. Käydään seuraavaksi läpi vastauksia hieman valikoiden.

Kyselyyn vastanneista kolmesta vastaajaa oli ilmoittanut, ettei heidän mielestään kaikki oppilaat tarvitse ohjelmoinnin opetusta. Näiden joukossa oli seuraavanlainen perustelu *"Tämän tarvitsevatko- kysymyksen voi esittää lähes jokaisesta peruskoulussa opetettavasta asiasta. Eiköhän asia ole niin, että kukaan ei tarvitse kaikkea, mitä peruskoulun opetussuunnitelmasta nykyisin löytyy. Joten vastaus on: "Ei tarvitse." "* Vastaaja halunnee korostaa sitä, etteivät kaikki tarvitse kaikkia yksityiskohtia mitä yläkoulussa opetetaan, mutta mitä opetussuunnitelma kuitenkin vaatii. Toinen poiminta vastauksista kuuluu seuraavasti *"Eivät tarvitse. Toisaalta ohjelmoinnin alkeet on ihan ok. Ohjelmointikielen/ohjelman valinta vaikuttaa ratkaisevasti. "* Äskeinen vastaus vaikuttaa jo hieman maltillisempi, sillä vastaaja korostaa ohjelmointikielen ja -ohjelmiston merkitystä opetuksessa. Viimeinen poiminta vastauksista: *"Eivät tarvitse, mutta ei siitä mitään haittaakaan ole. Peruskoulussa tarjotaan monessa muussakin asiassa joillekin oppilaille sellaista sisältöä, jota he eivät tule edes kohtuullisella tasolla koskaan saamaan haltuun. Esimerkiksi sin, cos ja tan opetetaan kaikille eikä niitä juuri kukaan ammattikouluun menevä käytä 9. luokan kokeen jälkeen."* Se mitä vastaaja mahdollisesti tarkoittaa on se, että peruskoulun opetussuunnitelmassa on mainittu taitoja, joiden opettaminen joillekin oppilaille ei ole tarpeellista. Kuitenkin vastaaja ilmaisee, että ohjelmoinnista opittavien taitojen opettamisesta ei ole kuitenkaan haittaa. Loput vastauksista olivat hyvin lyhyesti kielteisiä.

Vastausten joukosta löytyi myös seitsemän vastausta, joita (tutkielman) kirjoittaja ei pystynyt luokittelemaan mielekkäästi selkeästi kumpaankaan ääripäähän; myönteiseen tai kielteiseen. Käydään läpi muutama poiminta näistä puoliväliin jääneistä vastauksista. *"Ainakin algoritmisen ajattelun periaate tulisi olla hallinnassa."* ja *"Algoritminen ajattelu ja sen opettelu on tärkeää."* olivat vastauksina hyvin samankaltaisia, sillä molemmissa korostetaan algoritmisen ajattelun taitojen merkitystä. Seuraava vastaus korostanee myös kahden edellisen vastauksen ydinajatusta *"Ohjelmoinnin opetusta ei niinkään, mutta näen, että ohjelmistojen toimitusperiaatteiden tarkastelu tukisi matemaattisen, analyyttisen päätelyn opettamista: Mikä vaikuttaa mihinkin? Mitä tämän muuttaminen aiheuttaa?"*. Eräs vastaaja ilmaisi opetussuunnitelman velvoittavuuden, mutta kuitenkin myönsi, että ohjelmoinnin avulla voi oppia hyödyllisiä taitoja *"Ylhäältä määrätty niin sillä mennään. Ajattelun kehittymistä ainakin tarvitsevat ja siinä ohjelmointi on hyvästä."* Yksi vastaajista vastasi seuraavasti *"Eivät välttämättä, mutta yhtenäisessä peruskoulussa opetus tulee olla tasavertaista ja silloin kaikille kuuluu myös ohjelmoinnin opetus"*. Vastauksesta käy ilmi myös se, että ohjelmoinnin opetus oppilaiden tasavertaisuuteen vedoten on tärkeää. Viimeinen poiminta näistä puoliväliin jääneistä vastauksista kuuluu seuraavasti *"LibreOffice'n käytön oppiminen olisi tärkeää. Tulevaisuudessa ei ole välttämättä Microsoftin tuotteita muuten kuin pilvessä, eikä kaikilla ole siihen varaa."* Äskeinen vastaus korostaa

osaltaan sitä, että koulussa käytettävien oppimisympäristöjen ilmaisuus on tärkeä asia. Toisaalta vastaaja pitää tärkeänä, että oppilaat oppisivat perustoimistotyökalujen kuten kirjoitusohjelmiston, taulukkolaskentaohjelmiston ja muiden vastaavien ohjelmistojen peruskäytön hallintaa.

Aiemmista hieman lannistuneista vastauksista huolimatta kyselyn vastauksista yli puolet oli kuitenkin myönteisiä. Muutama vastaus korosti opetussuunnitelman velvoittavaa luonnetta. Vastaukset kuten "*Koska opetussuunnitelmaan on näin kirjoitettu, on valtakunnan taholta katsottu, että tarvitsevat. Olen kyllä itsekin sitä mieltä, että siitä voi olla monelle hyötyä, muttei se varmaan välttämätöntä olisi kaikille.*" ja "*Kyllä. Voiko muutoin vastata? Ops velvoittaa*" mainitsevat juurikin opetussuunnitelman velvoittavan roolin opetuksessa. Eräs vastaajista saattoi kokea kysymyksen turhaksi, sillä vastauksessa saattaa olla mukana hieman sarkasmia: "*Kyllä. Parasta olisi, kun kaikki maailman ihmiset oppisivat kaikki maailman tiedot ja taidot.*". Moni muu vastaaja korosti ohjelmoinnin perusteiden hallintaa; loogisen ajattelun kehittämistä; tietotekniikan perustaitojen opetusta; algoritmisen, ohjelmallisen tai ohjelmoinnillisen ajattelun kehittämistä. Näistä erityisesti algoritmisen ajattelun kehittämistä korostettiin, mikä saattaa olla juuri se, mitä opetussuunnitelmaa luotaessa ajateltiin. Kuten tässä luvussa on havaittu, tulkinnot ja mielipiteet tästä voivat poiketa hyvinkin merkittävästi toisistaan.

Seuraavaan opetussuunnitelmaan voisi olla paikallaan tarkennus siitä, mitä halutaan opettajien todella opettavan ohjelmoinnista. Halutaanko korostaa ohjelmallisen ja algoritmisen ajattelun roolia vai käyttää ihan konkreettisia ohjelmointikieliä osana opetusta. Vastausta pohtiessa herää kysymys: riittääkö uuden ohjelmointikielen opiskeluun aika? Aikaa täytyy varata itse ohjelmointikielen syntaksin oppimiseen, sillä ohjelmointikielet ovat kuten ihmiskielet: kielen oppiminen vie aikaa, eikä uuden kielen kirjoittaminen saati puhuminen suju keskiverto ihmiseltä vielä kahden kuukauden opiskelun jälkeen. Lisäksi ohjelmoinnissa tärkeitä on oppia kielen luova käyttö sekä itseilmaisussa että ongelmanratkaisussa.

4.2.7 Vastaajien huomioita ja kokemuksia

Tässä alaluvussa käydään läpi vastaajien kertomia huomioita ja kokemuksia ohjelmoinnin opetuksesta. Mikäpä olisikaan parempi verrata opetuksen toteutumisen onnistumista kuin kysyä itse opettajilta, kuinka he ovat kokeneet ohjelmoinnin opettamisen. Seuraavaksi käydään läpi mielenkiintoisimpia huomioita ja kokemuksia. Lukija voi itse pohtia, ovatko valinnat olleet mielekkäitä tai voisiko kokeilla niitä jopa itse.

Eräs vastaajista kirjoitti "*Toivon, että ohjelmointi saataisiin järkevämmiin osaksi matematiikan opetusta. Eikä näin, että pidetään ohjelmointiviikko ja sitten siirrytään tavalliseen matematiikkaan. Oppilailla ei ole taitoa työskennellä pitkäjänteisesti, joten ohjelmointi on monella haastavaa.*". Opettaja itsekin myöntää, että satunnaiset

ohjelmointiviikot eivät palvele tarkoitustaan, vaan paras olisi, mikäli ohjelmointi olisi integroituna paremmin matematiikan sisälle. Toisaalta vastaaja korostaa sitä, että nuorilla oppilailla on haasteita pitkäjänteisessä työskentelyssä, mitä ohjelmoinnin opiskelu ehdottomasti vaatii.

Toinen vastaaja vastasi seuraavasti: *"Kun ei ole tiedossa, mitä oikeasti pitäisi opettaa eikä kunnollisia oppimateriaaleja, on homma oikeasti vain leikkimistä. Valmiit materiaalit ovat tällä hetkellä todella vajavaisia tai sitten käytetty kieli on onneton (Racket). Lisäksi kestää viitisen vuotta, ennenkuin homma voi vakiintua, koska ensin pitää alakoulun puolella systeemin vakiintua, jotta yläkoulussa voidaan olettaa oppilailla olevan jotain taitoja aiheeseen liittyen."* Vastaajan huoli on ilmeinen, pysyvät muutokset vaativat pitkäjänteistä työtä ja merkittäviä muutoksia aiemmissa koulutusrakenteissa. Vastaaja huomauttaa, että muutoksen täytyy tulla alakoulun puolella, sillä yläkoululaisilla ei ole tarpeeksi tietoa tai taitoa ohjelmoinnista yläkouluun tullessaan. Vastaaja toivoisi myös selkeämpää ohjeistusta siitä, mitä oikeasti tulisi opettaa. Oppimateriaalien osalta vastaaja toivoo myös tapahtuvan kehitystä.

Kolmannella vastaajalla on melkoisesti kokemusta yliopistosta tietotekniikan osalta sekä ohjelmoinnista, sillä hän kertoo opiskelleensa noin 80 opintopistettä tietotekniikkaa, josta suuri osa ohjelmointia. Selvästi vastaajalla on vankka kokemus siitä, mitä ohjelmointi on sekä mahdollisesti siitä, mihin kaikkeen ohjelmointia voisi käyttää. Kuitenkin vastaajalla on ohjelmoinnin opetuksesta vähemmän kokemusta. Vastaaja ei kuitenkaan usko, että kaikki tarvitsevat ohjelmoinnin opetusta, *"... kaikilla ei ole tarpeeksi sinnikkyyttä ja algoritmiajattelun taitoa."*

Yhdellä vastaajalla oli hyvin selkeä tapa toteuttaa ohjelmoinnin opetusta. Tapaan sisältyy muuttuja, tekstin tulostaminen, laskutoimitukset, input-komennot, ehtorakenne, matematiikka-kirjaston käyttö ja silmukkarakenne. Opettaja on tehnyt ohjelmoinnin opetusta varten itse noin 20 sivuisen kokonaisuuden, jossa on teoriaa, esimerkkejä ja harjoitustehtäviä. Kyseinen opettaja on erittäin valveutunut ohjelmoinnin opetuksen osalta. Omien opetusmateriaalien tuottaminen vaatii opettajalta aina merkittävää omaa panostusta.

Eräs vastaajista kertoi mielipiteensä *"Ohjelmointi olisi parempi valinnaisaineena"*. Tämä voisi tietyissä tilanteissa paremmin kiinnostuneita oppilaita, mutta vuoden 2014 OPSissa on kuitenkin päätetty, että ohjelmointia tulisi kaikkien osata, joten tässä mielessä valinnaisuus ei ole vaihtoehto. Sama vastaaja jatkoi *"Monen oppilaan looginen päättelytaito on niin heikko, että yhdistettynä uuteen kieleen tehtävät ovat liian vaikeita ja eivät palvele motivaation puutteessa olevaa oppijaa mitenkään."* Tämä on oikein hyvä huomio, sillä mikäli oppilaan päättelytaidot ovat kaukana vaaditusta tasosta, uuden kielen opettelu voi olla ylitsepääsemättömän vaikeata. Tähän voi kuitenkin olla ratkaisuna ohjelmoinnin perusteiden tuleminen jo alakoulussa yksinkertaisilla ohjelmointikielillä.

Vaikka osa vastaajista ei kokenut kaikkien oppilaiden tarvitsevan ohjelmoinnin opetus-

ta, silti ohjelmallisen ajattelun opetus koetaan tärkeänä: *"Ohjelmoinnin opetusta ei niinkään, mutta näen, että ohjelmistojen toimintaperiaatteiden tarkastelu tukisi matemaattisen, analyyttisen päättelyn opettamista: Mikä vaikuttaa mihinkin? Mitä tämän muuttaminen aiheuttaa"*.

Eräällä vastaajalla oli mielenkiintoinen toteutustapa ohjelmointia opettaessaan, hän käytti kaksoistunnin jälkimmäisen tunnin ohjelmoinnin opettamiseen. Tämä tapa voi palvella hyvinkin mielekkäällä tavalla sekä oppilaita että opettajaa. Ensimmäinen tunti voisi olla ikään kuin teorian tunti ja uusien asioiden opettelua. Toinen tunti voisi olla enemmänkin soveltava tunti, jossa pääsisi parhaillaan käyttämään tietojaa soveltaen niitä erilaisissa pulmatehtävissä.

Yksi vastaaja koki eriyttämisen olevan ratkaisu ohjelmoinnin parempaan opettamiseen: *"Ohjelmoinnin opetus tulee vahvasti eriyttää. Algoritminen ajattelu softasta riippumatta on tärkeää kaikille, tämä onnistuu esim. graafisilla ohjelmointikielillä (Scratch). Innokkaimmille/lahjakkaimmille vahvasti soveltavia tehtäviä, mielellään niin että ohjelmoinnilla on joku käytännön sovellus. (Valinnainen aihe?)"*. Nykyisessä opetussuunnitelmassa ei kuitenkaan päädytty eriyttämiseen ohjelmoinnin osalta, vaan haluttiin kaikille opettaa algoritmista ja ohjelmallista ajattelua.

Moni matematiikan aineenopettaja saattaa jakaa seuraavan vastaajan mielipiteen: *"Haastavaa opettaa aihetta mihin ei ole itse perehtynyt kunnolla tai mistä ei ole itse kiinnostunut"*. Tällainen mielipide lienee täysin perusteltu, mikäli opettaja kokee, ettei hänellä ole tarpeeksi kattavaa aineenhallintaa kyseiseltä alalta, voi aineen opettaminen tuntua hyvinkin epämiellekkäältä. Kuten vastaaja itse kertoo, ei hänellä ole varsinaista mielenkiintoa ohjelmointia kohtaan. Tämä epäilemättä näkyy oppilaille, mikä tietysti huonommassa tapauksessa tarttuu oppilaisiin.

Luku 5

Pohdintaluku

Tämän luvun tavoitteena on kuvata tutkielman kirjoittajan omia ajatuksia aiheeseen liittyen, ja mahdollisesti herättää lukijaa pohtimaan ohjelmoinnin opetuksen tulevaisuutta Suomen kouluissa. Tätä tutkielmaa tehdessä tuli havaittua, että useita tutkielmia ja tutkimuksia on tehty siitä, miten ohjelmoinnin opettamista on toteutettu Suomen alakouluissa [Makkonen, J. & Pyykönen, A. (2018)] ja kuinka alakoulun opettajat ovat ottaneet vastaan ohjelmoinnin opetuksen [Karvonen, V-P. & Laukka, P. (2016)] ja [Raivonen, P. (2018)]. Opettajille on valmisteltu työkaluja ohjelmoinnin opettamiseen liittyen [Nurminen, T. (2017)]. Äskeiset ovat vain esimerkkejä tutkielmista ja lopputöistä, joita ohjelmoinnin opetukseen liittyen on tehty. Mainittakoon kuitenkin, että juurikaan yläkoulun puolelta vastaavanlaisia tutkielmia ei löytynyt. Tämä oli erityisenä motivaationa kirjoittaa tutkielma ohjelmoinnin opettamisesta matematiikan opetuksesta. Ohjelmoinnin oleminen uusi asia peruskoulun opetussuunnitelmassa, on perusteltua selvittää, kuinka opettajat ovat lähteneet mukaan näinkin merkittävän ja uuden asian tulemiseen koulumaailmaan.

Kyselyn tulosten valossa on ilmiselvää, että ohjelmoinnin opetus jakaa mielipiteitä matematiikan opettajien keskuudessa. Osa opettajista luultavasti harmittelee sitä, että matematiikan oppituntien lukumäärää ei lisätty, vaikka juuri lisättiin sisältöä matematiikan sisälle. Tämä muutos yläkoulussa voi tuntua huomattavan suurelta, sillä osalle oppilaista nykyinenkään aika ei tahdo riittää pelkän matematiikan osalta. Puhumattakaan uuden kielen ja loogisen ajattelun opiskelussa, mitä syvimmiltään ohjelmoinnin opettelu yläkoulussa on. Moni opettaja saattaa kokea tämän ikävänä epäsuhtana.

Kuten luvun 4 alussa huomattiin, on eri työkokemusryhmien välillä eroja siinä, osallistuvatko he koulutuksiin vai eivät. Syitä tähän ei kuitenkaan tiedetä. Mahdollisesti ero johtuu turtumisesta muutosten määrään, mitä kyseiset opettajat ovat kohdanneet opetusuransa aikana. Vähän kokemusta omaavat opettajat saattavat myös kokea velvollisuudekseen tuoreina opettajina käydä koulutuksissa. Vähän kokeneet opettajat voivat olla

myös aidosti kiinnostuneita ja uteliaita siitä, kuinka toteuttaa ohjelmoinnin opeusta.

5.1 Jatkotutkimusaiheita

Tätä tutkielmaa tehdessä heräsi muutama kysymys, mihin valitettavasti tätä tutkielmaa varten tehty kyselytutkimus ei anna vastausta. Esimerkiksi se, miten maantieteellinen sijainti olisi vaikuttanut näihin tuloksiin. Olisi mielekiintoista kuulla, onko Suomessa erityisesti joitakin alueita, jotka kärsivät resurssipulasta enemmänkin. Onko joillakin alueilla opettajien vaikeampaa saada koulutusta ohjelmoinnin opetukseen liittyen?

Toisaalta suuremman otannan saaminen vastaavalla kyselyllä voisi antaa myös lisäinformaatiota siitä, miten yleisiä ääripäät monessakin kysymyksessä ovat. Onko jakauma lähes identtinen tämän kyselyn jakauman kanssa, vai olisiko havaittavissa merkittäväkin ero näiden kahden välillä.

Kyselylomakkeen kysymyksien valitseminen ja muotoileminen vaati paljon aikaa, sillä tavoitteena kyselyissä on juurikin selvittää aiemmin ennaltapäätettyä asiaa. Vaikka aiemmin tässä tutkielmassa harmiteltiin sitä, että opettajien sijaintia ei tultu selvittäneeksi. Toisaalta tätä voidaan pitää hyvänä asiana vastaajien anonymiteetin säilyttämisen kannalta. Mikäli vastaajan työpaikan (koulun) sijainti olisi saatu selville, ei tällöin monissa kouluissa olisi ollut mahdollisuutta vastata tähän kyselyyn pelkäämättä vastaajan anonymiteetin menettämisen puolesta. Mikäli oltaisiin kysytty vastaajan työpaikan (koulun) sijaintia kaupungin tai kunnan tarkkuudella, olisi tällöin ollut myös riski menettää anonymiteetti. Viimeistään silloin, jos olisi yhdistetty vastaajan työkokemus vuosina, olisi menetetty vastaajan anonymiteetti. Mielekästä olisi ollut kysyä vastaajan koulun sijaintia maakunnan tarkkuudella. Tällöin ei olisi ollut mielekästä pelätä anonymiteetin menettämisen puolesta, vaikka työkokemus vuosina oltaisiin yhdistetty sijaintiin maakunnan tarkkuudella.

Erityisen kiinnostavaa olisi ollut selvittää, minkälaisia erilaisia projekteja ja oppimiskokonaisuuksia opettajat ovat kehittäneet yhteistyössä muiden aineiden opettajien kanssa. Matematiikan ja fysiikan opettajien yhteistyön hedelmänä voi tulla hyvinkin mielenkiintoisia projekteja, esimerkkejä näistä voisi olla omien robottien tekeminen ja mikroelektronikan yhdistäminen fysiikan ja ohjelmoinnin kautta. Erilaisten sensorien, kuten lämpötilasensorien, kosketussensorien ja infrapunasensorien käyttö fysiikan oppitunneilla voisi olla oppilaiden mielestä hyvinkin innostavaa. Äskeiset esimerkit olivat vain hyvin ilmeisimmistä päästä. Koska pitkissä suomenkielisissä ja englanninkielisissä tekstissä on eri vokaalisuhde, voitaisiin ädinkielessäkin esimerkiksi tutkia sitä, kuinka tietokone voi tunnistaa kirjoittaako käyttäjä suomea vai englantia. Vaihtoehdot ovat rajattomat.

Vaikka uusimmassa opetussuunnitelmassa on vain vähän täsmällisiä mainintoja ja tuskin yhtään yksikäsitteistä neuvoa siitä, miten ohjelmoinnin opetusta pitäisi toteuttaa, on

tämä opettajien kannalta ikään kuin kaksiteräinen miekka. Toisaalta monet opettajat arvostavat suuresti sitä, että he voivat toteuttaa ohjelmoinnin opetusta juuri siten, kuin he parhaaksi näkevät. Toisaalta taas monet opettajat toivoisivat hieman tarkempaa ohjeistusta siitä, kuinka he voisivat toteuttaa ohjelmoinnin opetusta ylipäättään. Tietysti opetussuunnitelmaa laativien henkilöiden tulee pitää mielessä se, että mikäli tarkka ohjeistus annettaisiin, ei tämä tietty tapa välttämättä sopisi kaikille. Kuten eräs kyselylomakkeeseen vastannut ilmoitti, että hän kokee ohjelmoinnin opettamisen haastavana ja mielenkiinnottomana, olisikin mielenkiintoista selvittää, kuinka monta opettajaa jakaa kyseisen vastaajan mielipiteen. Eräs syy tähän voikin olla juuri se, että osalle matematiikan opettajista tuli työtehtävikseen opettaa aiheita, jota ei ole aiemmin opiskellut. Sekä äskeisen lisäksi se, että työtunteja ei tullut tähän lisää, vaan opetus tulisi toteuttaa samassa ajassa kuin aiemmin, mutta silti monipuolisemmin aiheista, joita ei välttämättä itse osaa.

Kyselyyn osallistujat kerättiin pääosin sosiaalisen median kautta ja matemaattisten aineiden opettajien liiton MAOL ry:n sähköpostimuodossa kulkevan "kirjeen" avulla. Tällä tavoin saatetaan saada tietynlaisia vastaajia mukaan, kuten esimerkiksi hyvin aktiiviset opettajat. Sosiaalisen median kautta linkki kyselyyn pystyttiin jakamaan äärimmäisen helposti kyselystä kiinnostuneille. Lisäksi tavoitettiin erittäin huomattava määrä potentiaalisia vastaajia. Vielä kaksikymmentä vuotta sitten tämänlainen vastaajien lähestyminen ei olisi ollut mahdollista. Tämän sijaan olisi tullut tehdä lehti-ilmoitus useissa eri lehdissä, eikä tällöinkään olisi välttämättä tavoitettu yhtä hyvin vastaajia kuin sosiaalisen median avulla. Toinen vaihtoehto kyselylle olisi ollut käydä koulu kerrallaan pyytämässä vastaajia osallistumaan kyselyyn. Tämä olisi ollut äärimmäisen työläs lähestymistapa, eikä tämän tutkielman keston kannalta todettu järkeväksi tavaksi. Kuitenkin vankempi vastaajakanta oltaisiin luultavasti tavoitettu tällä tavoin menettelemällä.

Huolestuttavin asia, mikä kyselyn vastauksia analysoidessa tuli vastaan oli se, että osa opettajista ei aidosti koe omaavansa tarpeeksi resursseja toteuttaa ohjelmoinnin opetusta siten, kuin on tarkoitettu. Tämä saattaa oppilaat eri puolilla Suomea eriarvoiseen asemaan, eikä tämä ole opetussuunnitelman perusteiden saati perusopetuslain tavoitteiden mukaista. Mikäli opettajat eivät koe, että heillä on tarpeeksi tietokoneita ohjelmoinnin opettamisen toteuttamiseen on tämä huolestuttavaa. Tietokoneen tarjoaminen oppilaalle koulun puolesta on myös melko suuri vaatimus kunnille, joilla on vähemmän rahaa. Kuitenkin ohjelmoinnin oppimiseen tarvitaan tietokonetta, kuten tässäkin tutkielmassa on aiemmin todettu. Ne oppilaat, jotka suuntaavat opinpolkunsä lukioon, tulevat tarvitsemaan tietokonetta myös toisen asteen opinnoissa, sillä nykyään myös ylioppilastutkinto suoritetaan sähköisesti [Ylioppilastutkinto].

Monet opettajat eivät kokeneet omaavansa tarpeeksi oppimateriaaleja ohjelmoinnin opetusta toteuttaessaan. Tämä korjaantunee tietoisuuden lisääntymisen ja ajan myötä, kun parempia oppimateriaaleja kehitellään jatkuvasti, sekä tietoisuus näistä parantunee. On silti täysin ymmärrettävää, että opettajat eivät koe pystyvänsä opettamaan ohjel-

mointia halutulla tasolla, mikäli heillä ei ole työkaluja siihen.

Viimeisin resurssi, mistä kyselyssä oltiin kiinnostuneita oli aika. Niinkin moniulotteinen asia kuin ohjelmointi, vaatii runsaasti aikaa taitojen oppimiseen. Onkin huolestuttavaa, että opettajat eivät ole kokeneet, että heillä olisi ollut tarpeeksi aikaa opettaa ohjelmointia matematiikan opetuksessa. Yksi mahdollinen ratkaisu tähän olis se, että opituntien lukumäärää lisättäisiin matematiikkaan, jotta opettajilla olisi tarpeeksi aikaa toteuttaa heille määrätty tehtävät hyvin. Mikäli aikaa ei lisätä, toinen vaihtoehto on se, että jostain muusta kokonaisuudesta jätettäisiin yksityiskohtia pois, tai vaihtoehtoisesti jotkin aiheet jätettäisiin kokonaan käsittelemättä.

Luku 6

Johtopäätösluku

Tutkielman alussa käytiin läpi, miten ohjelmointi näkyy uusimman opetussuunnitelman perusteissa yläkoulussa. Ohjelmointi on nykyään osa matematiikan opetusta ja sisältää vahvasti yhteyksiä laaja-alaisen osaamisen kokonaisuuksiin. Ohjelmoinnin avulla opittavat taidot koetaan nykypäivänä erityisen tärkeäksi taidoksi, jotka kaikilla tulisi olla. Oppilaiden tasa-arvoisten oppimismahdollisuuksien takia, on ensiarvoisen tärkeää, että kaikki oppilaat saavat opetussuunnitelman mukaista opetusta ja oppisivat näitä taitoja.

Ohjelmoinnista ja ohjelmoinnin opettamisesta puhuttaessa törmätään erilaisiin termeihin, kuten ohjelmoiminen, koodaaminen, algoritmien ajattelu ja CT. Näiden määritelmät käytiin tutkielmassa läpi yksi kerrallaan ja selvennettiin hieman hämmennystä, joka näihin termeihin liittyy. Ohjelmointi on sekä ongelman jakamista osiin että luovaa ongelmanratkaisua kun taas koodaaminen on pelkkää lähdekoodin kirjoittamista ilman sen syvällisempää ymmärrystä. Algoritmien ajattelu ja CT ovat melko lähellä toisiaan, sillä molemmat kuvaavat ohjelmoinnin luovan puolen hahmottamisen ja toteuttamisen oppimista. Tässä tutkielmassa näiden kahden termin eroja kuitenkin yritettiin selventää. Siinä missä algoritmisessa ajattelussa tarkennetaan ongelman ratkaisemisen vaiheita ja jo annettujen vaiheiden parantamista, kuvaa CT enemmänkin ajatteluprosessia hyvin yleisellä tasolla. Näiden jälkeen kerrottiin kuinka ohjelmointia, algoritmista ajattelua ja CT:tä opetetaan eri puolilla maailmaa. Tutkielmassa käytiin läpi erilaisia toteutuksia ympäri Eurooppaa ja samalla havaittiin, että käytännöt ovat hyvin kirjavat.

Tutkimusmenetelmänä käytettiin kyselytutkimusta, jonka kysymykset muotoiltiin ja valittiin suurella huolella. Kyselytutkimus koettiin luontevaksi tavaksi selvittää opettajien mielipiteitä ja kokemuksia erinäisistä ohjelmointiin ja ohjelmoinnin opettamiseen liittyvistä aiheista. Kyselyssä selvitettiin muun muassa opettajien taustoja ja asenteita, opettajien resursseja, ohjelmoinnin opettamisen toteuttamisesta ja mielipiteitä ohjelmoinnin opettamisesta, ja lopuksi opettajan omaa ohjelmointitaitoa ja kouluttautumista ohjelmointiin liittyen. Kyselyn lopussa vastaajat saivat kommentoida omia mietteitään vapaa

sana -osioon.

Tutkimusaineisto koostui 46 matemaattisten aineiden opettajasta, jotka opettavat ainakin matematiikkaa yläkoulussa. Tuloksia analysoitiin sekä kvalitatiivisesti että kvantitatiivisesti. Kvalitatiivisesti tuloksia tulkittiin vastausten luokittelulla ja tyypittelyllä [Tuomi, J. & Sarajärvi, A. (2009)]. Kvantitatiivisesti vastauksia tutkittiin Kruskal-Wallis H-testillä sijaluvuille. Testiä tehdessä käytettiin alfan arvoa 0,05, jota käyttämällä saatiin 95% varmuudella oikea tulos. Ennen testin tekemistä tehtiin kuitenkin nollahypoteesi, joka oli: "Sekä kokeneemmat että vähemmän kokeneet opettajat ovat osallistuneet yhtä aktiivisesti koulutukseen ohjelmoinnin opetukseen liittyen.". Lisäksi annettiin vaihtoehtoinen hypoteesi, eli "Näiden ryhmien välillä on eroa.". Testin tuloksista kävi ilmi, että nollahypoteesi ei päde, joten vaihtoehtoinen hypoteesi kuvaa todellisuutta paremmin, eli testin mukaan ryhmien välillä on eroa.

Testin jälkeen käytiin kyselyn tuloksia muutamista mielenkiintoisimmiksi valituista kysymyksistä kuten se, toteuttavatko opettajat ohjelmoinnin opetusta matematiikassa vai erikseen valinnaisena kurssina. Lisäksi käytiin mm. läpi tapoja toteuttaa ohjelmoinnin opetusta matematiikan opetuksessa, sekä onko opettajilla tarpeeksi resursseja, kuten tietokoneita, oppimateriaaleja ja aikaa, toteuttaessaa ohjelmoinnin opetusta. Seuraavaksi käytiin muutamia sanallisia vastauksia siitä, tarvitsevatko kaikki oppilaat ohjelmoinnin opetusta. Lopuksi käytiin läpi joitakin vastaajien yleisiä huomioita ja kokemuksia.

Tutkielman lopussa olevassa pohdintaluvussa pyrittiin perustelemaan ja pohtimaan kuinka onnistunut tutkielma oli sekä se, miten sitä olisi voinut parantaa ja mitä olisi voinut tehdä eri lailla. Tämän lisäksi mainittiin muutamia mahdollisia ja kirjoittajan mielestä mielenkiintoisia jatkotutkimusaiheita ja -ideoita.

Luku 7

Liitteet

Kyselytutkimus osa 1.

Olen

- * Kysymys 1. ☐ Mies.
☐ Nainen.
☐ Muu/En halua vastata.

Työkokemus vuosina matematiikan opettajana

- * Kysymys 2. ☐ 0 - 5 vuotta.
☐ 6 - 15 vuotta.
☐ 16 - 29 vuotta.
☐ Yli 30 vuotta.

Omaa asennettani ohjelmointia kohtaan kuvaa parhaiten:

- * Kysymys 3. ☐ En ole lainkaan kiinnostunut.
☐ En ole juurikaan kiinnostunut.
☐ Olen jossain määrin kiinnostunut.
☐ Olen hyvin kiinnostunut.
☐ Olen erittäin kiinnostunut.

Omaa asennettani ohjelmoinnin opetusta kohtaan kuvaa parhaiten:

- * Kysymys 4. ☐ En ole lainkaan kiinnostunut.
☐ En ole juurikaan kiinnostunut.
☐ Olen jossain määrin kiinnostunut.
☐ Olen hyvin kiinnostunut.
☐ Olen erittäin kiinnostunut.

Miten ohjelmoinnin opetusta matematiikassa pitäisi toteuttaa?

- * Kysymys 5. ☐ Microsoft Excelin / LibreOffice Calcin opettaminen riittää.
☐ Oppilaiden pitäisi oppia algoritmista/ohjelmallista ajattelua.
☐ Oppilaiden pitäisi oppia kirjoittamaan yksinkertaista koodia.
☐ Muu (Miten?).

Jos vastasit äskeiseen "Muu",
voit tarkentaa tähän
kenttään.

* Kysymys 6. Tarvitsevatko kaikki oppilaat ohjelmoinnin opetusta?

Kyselytutkimus osa 2.

* Kysymys 7. Miten olet toteuttanut ohjelmointia matematiikan opetuksessa?

* Kysymys 8. Miten olet ajoittanut ohjelmoinnin matematiikan opetuksessa?

Miten olet toteuttanut ohjelmointia matematiikan opetuksessa?

- * Kysymys 9. ☐ Kynällä ja paperilla.
☐ Tietokoneella.
☐ Muulla tavalla, miten?

Jos vastasit äskeiseen
"Muulla tavalla", voit
tarkentaa tähän kenttään.

Kysymys 10. Kuinka paljon sinulla on ollut seuraavia resursseja käytössäsi ohjelmoinnin opetusta toteuttaessasi matematiikassa:

a. Tietokoneita

- * ☒ Ei ollenkaan.
☐ Liian vähän.
☐ Melkein riittävästi.
☐ Riittävästi.

b. Oppimateriaaleja

- * ☒ Ei ollenkaan.
☐ Liian vähän.
☐ Melkein riittävästi.
☐ Riittävästi.

c. Aikaa

- * ☒ Ei ollenkaan.
☐ Liian vähän.
☐ Melkein riittävästi.
☐ Riittävästi.

Oletko tyytyväinen siihen, miten ohjelmointia toteutetaan matematiikan opetuksessa tällä hetkellä omassa koulussanne?

- * Kysymys 11. ☒ En ole ollenkaan tyytyväinen nykyiseen toteutustapaan, enkä tiedä miten sitä voisi parantaa.
☐ En ole tyytyväinen, vaikkakin mielestäni sitä voitaisiin toteuttaa paremmin.
☐ Olen tyytyväinen, vaikkakin mielestäni sitä voitaisiin toteuttaa paremmin.
☐ Olen erittäin tyytyväinen, eikä mielestäni sitä ole tarve toteuttaa paremmin.

Oletko toteuttanut ohjelmointia muissa aineissa (esimerkiksi fysiikassa) yhteistyönä?

- * Kysymys 12. ☐ En.
☐ Kyllä (fysiikassa).
☐ Kyllä (muissa aineissa, missä?).

Jos vastasit äskeiseen "Kyllä", voit tarkentaa tähän kenttään.

Onko oppilailla ollut keskimäärin riittävät esitiedot ohjelmoinnista?

- * Kysymys 13. ☐ Ei.
☐ Kyllä.

Onko sinulla ollut käytössäsi hyviä oppimateriaaleja ohjelmoinnin opetuksessa?

- * Kysymys 14. ☐ Ei.
☐ Kyllä (Mikä/Mitkä?).

Jos vastasit äskeiseen "Kyllä", voit tarkentaa tähän kenttään.

Kysymys 15. Millä ohjelmointikielellä olet toteuttanut ohjelmointia? (Valitse vähintään yksi vaihtoehto, voit valita useita)

- ☐ Pythonilla.
☐ Javalla.
☐ JavaScriptillä.
☐ C++ :lla.
☐ Jollain muulla.
☐ En millään.

Onko koulussanne toteutettu ohjelmoinnin opetus matematiikassa vai pidetäänkö kurssi erikseen?

- * Kysymys 16. ☐ Ohjelmoinnin opetus on toteutettu matematiikassa.
☐ Kurssi pidetään erikseen.
☐ Muu (Miten?).
☐ Ei ole toteutettu.

Jos vastasit äskeiseen "Muu", voit tarkentaa tähän kenttään.

Kyselytutkimus osa 3.

Oletko opiskellut itse ohjelmointia ja/tai ohjelmoinnin pedagogiikkaa?

- * Kysymys 17.
- ☐ En ole opiskellut kumpaakaan.
 - ☐ Olen opiskellut ohjelmointia.
 - ☐ Olen opiskellut ohjelmoinnin pedagogiikkaa.
 - ☐ Olen opiskellut sekä ohjelmointia että ohjelmoinnin pedagogiikkaan.

Kuinka paljon sinulla on ohjelmointikokemusta?

- * Kysymys 18.
- ☐ Ei yhtään.
 - ☐ Olen nähnyt lyhyitä ja yksinkertaisia koodinpätkiä matematiikan oppikirjassa.
 - ☐ Olen silmäillyt materiaaleja netistä, kuten esimerkiksi Code.org tai ohjelmointikurssi.github.io.
 - ☐ Olen kirjoittanut itse muutaman rivin koodia.
 - ☐ Olen harrastanut ohjelmointia vapaa-ajallani.

Oletko osallistunut koulutuksiin ohjelmoinnin opetukseen liittyen?

- * Kysymys 19.
- ☐ Olen osallistunut oman kouluni järjestämään koulutukseen.
 - ☐ Olen osallistunut muun tahon puolesta (esimerkiksi MAOL-päivät) järjestämään koulutukseen.
 - ☐ Muu (Mihin?).

Jos vastasit äskeiseen "Muu",
voit tarkentaa tähän
kenttään.

Vapaa sana

Vapaa sana

Lähteet

- [Bennedsen, J., ym. (2008)] Reflections on the Teaching of Programming, Methods and Implementations. Print ISBN 978-3-540-77933-9. s. III - IV (esipuhe), s. VII - VIII (johdanto), s. 17 - 67. Saatavilla: <https://doi-org.libproxy.helsinki.fi/10.1007/978-3-540-77934-6>
- [code.org] code.org sivustolta esimerkkitehtävä ohjelmointipelistä <https://studio.code.org/hoc/1>
- [El Kamel, S. (2018)] El Kamel, S. (2018). Mitä se koodaaminen ihan oikeasti on? HS:n animaatiot näyttävät, miten syntyy todella helppo painoindeksilaskuri. Julkaistu Helsingin Sanomissa 10.7.2018. Saatavilla: <https://www.hs.fi/teknologia/art-2000005749972.html>
- [European Schoolnet (2015)] European Schoolnet & Balanskat, A. & Engelhardt, K. (2015). Computing our future: Computer programming and coding, Priorities, school curricula and initiatives across Europe. s. 13 - 18, s. 21 - 22, s. 46 - 55, s. 62 - 70. Saatavilla: http://fcl.eun.org/documents/10180/14689/Computing+our+future_final.pdf/746e36b1-e1a6-4bf1-8105-ea27c0d2bbe0
- [Futschek, G.(2006)] Futschek, G. (2006). Algorithmic Thinking: The Key for Understanding Computer Science. s. 159 - 161, s. 168.
Saatavilla: https://www.researchgate.net/profile/Gerald_Futschek/publication/221437678_Algorithmic_Thinking_The_Key_for_Understanding_Computer_Science/links/54edfdc50cf2e2830863ce05/Algorithmic-Thinking-The-Key-for-Understanding-Computer-Science.pdf?origin=publication_detail
- [Karvonen, V-P. & Laukka, P. (2016)] Karvonen,V-P., Laukka, P. (2016). Suomalaisten opettajien asenteita ja valmiuksia ohjelmoinnin opetukseen, Pro Gradu. Saatavilla: <http://urn.fi/URN:NBN:fi:oulu-201602121186>

- [Kekäläinen, O. (2015)] Kekäläinen, O. (2015). Onko automatisointiajattelu paras suomenkos käsitteestä "computational thinking"? Tuovi 13: Interaktiivinen tekniikka koulutuksessa 2015-konferenssin tutkijatapaamisen artikkelit (toim. Viteli, J. & Östman, A.) (s. 27 - 39). ISBN 978-951-44-9909-8 Saatavilla: <http://urn.fi/URN:ISBN:978-951-44-9909-8>
- [Kruskal, W. & Wallis, A.W. (1952)] Kruskal, W. & Wallis, A.W. (1952). "Use of ranks in one-criterion variance analysis". Journal of the American Statistical Association. 47 (260): s. 583 – 621. doi:10.1080/01621459.1952.10483441.
- [Kyselylomake] Kyselylomake ohjelmoinnin opetuksesta matematiikan opetuksessa yläkoulussa, laatinut Jaakko Kokkonen (avustanut Sirkka-Liisa Eriksson). Liitteenä.
- [Makkonen, J. & Pyykönen, A. (2018)] Makkonen, J., & Pyykönen, A. (2018). "Se on mun mielest taas yks tapa rikastuttaa sitä opiskelua ja oppimista, itekki oppii sit uusia juttui."– Alakoulun opettajien käsityksiä ohjelmoinnin opettamisesta Saatavilla: <https://www.innokas.fi/wp-content/uploads/2018/08/Jenni-Makkonen-ja-Anniina-Pyykkoenen-Gradu.pdf>
- [Mannila, L. ym. (2014)] Mannila, L., Dagiene, V., Demo, B., Grgurina, N., Mirolo, C., Rolands, L., Settle, A. (2014). Computational thinking in K-9 Education. s. 1 - 9. Saatavilla: https://www.researchgate.net/publication/273772180_Computational_Thinking_in_K-9_Education
- [Netbeans] Netbeans ohjelmointiympäristön kotisivut, <https://netbeans.org/>
- [Nordén ym.] Nordén, L-Å., Heintz, F., Mannila, L., Parnes, P., Regnell, B. (2017) Introducing Programming and Digital Competence in Swedish K-9 Education. s. I - II. Saatavissa: <http://urn.kb.se/resolve?urn=urn:nbn:se:uu:diva-337182>
- [Nurminen, T. (2017)] Nurminen, T., (2017). Lasten koodauksen opetus - Työkaluja opettajille Saatavilla: <https://dspace.cc.tut.fi/dpub/bitstream/handle/123456789/25013/Nurminen.pdf?sequence=4&isAllowed=y>
- [Ohjelmointikurssi.github.io] Ohjelmoinnin harjoitteluun soveltuva sivusto JavaScript-ohjelmointikielellä. <https://ohjelmointikurssi.github.io/>
- [Opetus- ja kulttuuriministeriö] Opetus- ja kulttuuriministeriö, Peruskoulujen rahoitus Suomessa: <https://minedu.fi/rahoitus>
- [OPS 2004] Opetushallitus (2004), Perusopetuksen opetussuunnitelman perusteet 2004, luettu 8.1.2019, ISBN952-13-1992-5 (pdf) Saatavilla: https://www.oph.fi/download/139848_pops_web.pdf

- [OPS 2014] Opetushallitus (2014), Perusopetuksen opetussuunnitelman perusteet 2014, luettu 8.1.2019, ISBN 978-952-13-5999-6 (pdf) Saatavissa: https://www.oph.fi/download/163777_perusopetuksen_opetussuunnitelman_perusteet_2014.pdf
- [Orponen, P.] Orponen, P. (2003) Tietotekniikkaan ennen tietokoneita: Alan Turing ja tietotekniikan kiehtovat alkuvaiheet. Tietojenkäsittelyteorian laboratorio: Teknillinen korkeakoulu Espoo Dipoli Saatavilla: http://users.ics.aalto.fi/orponen/papers/itvhist_2003.pdf
- [Raivonen, P. (2018)] Raivonen, P. (2018). "Rohkeasti vain kokeilemaan" Tutkimus alakoulun opettajien valmiudesta opettaa ohjelmointia. Saatavilla: https://helda.helsinki.fi/bitstream/handle/10138/233147/PetraRaivonen_Gradu.pdf?sequence=2&isAllowed=y
- [Ristiintaulukointi] Ristiintaulukointi <https://www.fsd.uta.fi/menetelmaopetus/ristiintaulukointi/ristiintaulukointi.html>
- [Seppänen, R. ym. (2005)] Seppänen, R., Kervinen, M., Parkkila, I., Karkela, L., Meriläinen, P. MAOL-työkalut : matematiikka, fysiikka, kemia. 2.-8. painos., s. 61. ISBN-13: 978-951-1-20607-1
- [Tie.koodariksi.fi] Ohjelmoinnin harjoitteluun soveltuva sivusto Python-ohjelmointikielellä. <https://tie.koodariksi.fi/>
- [Tuomi, J. & Sarajärvi, A. (2009)] Tuomi, J. & Sarajärvi, A (2009), Laadullinen tutkimus ja sisällönanalyysi, ISBN 978-951-31-4865-2, 5. uudistettu laitos. s. 91 - 104.
- [Vee, A. (2013)] Vee, A. (2013). Understanding Computer Programming as a Literacy. Literacy in Composition Studies Journal, Vol1, No2 (2013). s. 44 - 53 Saatavilla: <http://licsjournal.org/OJS/index.php/LiCS/article/view/24/26>
- [Wikipedia Computer Programming] (Tietokone)ohjelmoinnin määritelmä, luettu 5.7.2019 https://en.wikipedia.org/wiki/Computer_programming
- [Ylioppilastutkinto] Ylioppilastutkintolautakunnan kotisivut, digitaalinen ylioppilastutkinto. <https://www.ylioppilastutkinto.fi/ylioppilastutkinto/digitaalinen-ylioppilastutkinto>
- [Wikipedia HAVO] HAVO koulutus https://en.wikipedia.org/wiki/Hoger_algemeen_voortgezet_onderwijs
- [Wikipedia VWO] VWO koulutus https://en.wikipedia.org/wiki/Voorbereidend_wetenschappelijk_onderwijs